EXHIBIT 2

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT COURT OF NEW YORK

---x
KMS TECH, INC.,

Plaintiff,

Civil No.: 1:20-cv-01041-GBD-DCF

-against-

PLAINTIFF KMS TECH INC.'S
<u>EXPERT WITNESS DESIGNATION</u>

G MISSION INC., *et al.*,

Defendants
---x

Pursuant to Federal Rule of Civil Procedure, Rule 26(a)(2), Plaintiff KMS Tech, Inc. ("Plaintiff") designates Kenneth J. Amron ("Mr. Amron"), 545 West End Avenue, #3F, New York, NY 10024, as a retained expert witness in the above-entitled matter. Attached hereto and incorporated herein as **<u>Exhibit 1</u>** is Mr. Amron's report.

Mr. Amron has more than thirty years of experience in the software industry and is currently a practicing forensic consultant experienced with software and hardware interfaces and designing, coding, deploying and operating software that processes and integrates with digital media, including audio, video and imagery.

Mr. Amron's compensation for his work in preparing his accompanying report is $350/hour. The rate for testifying in depositions or trial and preparing for such activities is $450/hour. Mr. Amron's compensation was not and is not contingent upon the substance of his opinions or testimony, or the outcome of this case.

Plaintiff reserves the right to have Mr. Amron supplement his report as necessary or appropriate, and/or to provide a rebuttal report should there be a need. Plaintiff reserves the right to seek and present direct and rebuttal testimony from Mr. Amron based upon all evidence, disclosed materials in this matter, and testimony at trial.

Hard copy disclosure of the accompanying report and its incorporated and referenced materials on an accompanying USB drive was made on October 22, 2024, via FedEx, and confirmed as received on October 23, 2024.

Dated: October 25, 2024
New York, New York

Sincerely,

PARDALIS & NOHAVICKA, LLP

By: /s/Eleni Melekou
Eleni Melekou, Esq.
950 Third Avenue, 11th Floor
New York, NY 10022
(212) 213-8511
Eleni@pnlawyers.com

Attorneys for Plaintiff

EXHIBIT 1

KMS TECH, INC

Plaintiff,

vs.

G. MISSION, INC.,

Defendant.

Civil Action No. 1:20-cv-01041
District Court, S.D. New York

OPENING EXPERT REPORT OF KENNETH J. AMRON

Contents

Table of Figures

I. INTRODUCTION

1. I, Kenneth Amron, have been retained by KMS Tech, Inc. ("**KMSTech**") counsel as a consultant and expert in this action. I was tasked to visit a location identified to me as G. Mission, Inc's ("**GMission**") premises, to review and obtain computer code "from all computers at that location" for the purpose of comparison to KMSTech's Copyright Deposit ("**p_deposit**").

2. This report describes the visit and my findings regarding the comparison I performed between the computer code found at the Defendant's premises and the computer code of the Plaintiff's Deposit.

II. PROFESSIONAL BACKGROUND AND QUALIFICATIONS

3. I am a currently a forensic consultant and have over thirty years of experience in the software industry. My focus in a variety of roles has included digital media in both embedded, distributed and Internet configurations. A copy of my CV is attached as Exhibit 1.

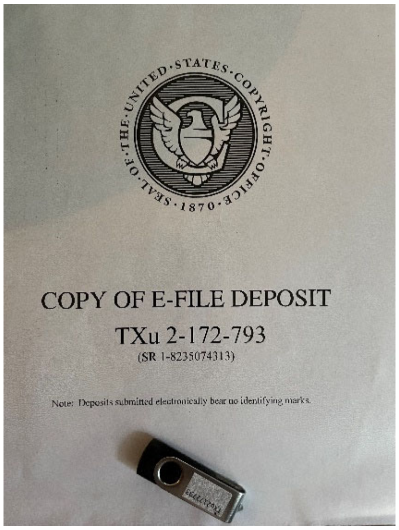
4. Over the course of my career, I have been responsible for software and hardware interfaces and designing, coding, deploying and operating software that processes and integrates with digital media, including audio, video and imagery.

III. COMPENSATION.

5. My compensation is not dependent on the substance of my opinions or my testimony or the outcome of this case. I have no financial interest in the outcome of this case.

IV. MATERIALS CONSIDERED

6. In connection with this matter, I have reviewed a USB stick identified as TXu 2-172-793 received with a coversheet containing the "SEAL OF THE UNITIED STATES COPYRIGHT OFFICE", in a sealed plastic, the KMS Tech Copyright deposit, ("**p_deposit**")[1], provided to me by KMSTech counsel.



COPY OF E-FILE DEPOSIT

TXu 2-172-793

(SR 1-8235074313)

Note: Deposits submitted electronically bear no identifying marks.

Figure IV-1 - Image of received copyright material

7. I have reviewed the deposition of Defendant, Mr. H. Kwak.

8. I have reviewed the deposition of Plaintiff's, developer, Mr. Songjae Han.

9. I have reviewed the deposition of Plaintiff, Mr. Yun Sue An.

10. I have reviewed the court order regarding discovery, ECF 158.

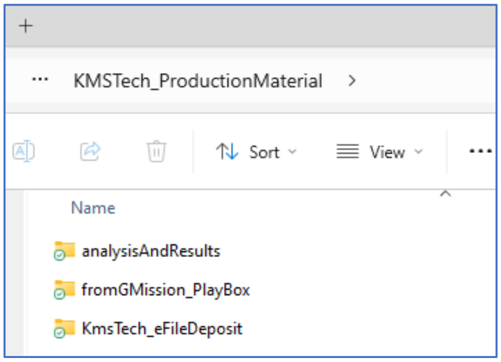
11. I have reviewed and subjected to analysis material copied from a Playbox[2] system, [*see infra VII, "*VISIT TO GMISSION*"*], specifically files found on the Playbox system at the Defendant's premises, precisely, files on that system's "E: drive", referred to herein as "**d_playbox**" [*see infra,* ¶*21.a*].

[1] The three file names found on the Deposit are:
(i) 1-8235074313-20191219-165336-0-6081967-F7541-playbox_souece_code.txt
(ii) 1-8235074313-20191219-165336-1-6081992-42320-kmspos_souece_code.txt
(iii) 1-8235074313-20191219-165336-2-6081995-B3B01-ssmedia_lite_manager_source_code.txt
These files were written in the Microsoft C# programming language.

[2] The court order [*See, ECF 158*] provided for my access to all versions of Plaintiff's software in Defendant's possession. During my visit to the Defendant's premises, I was not granted permission to view the machines that were in the Karaoke rooms or any other system at the Defendant's premises. [*See, infra, Section VII,* VISIT TO GMISSION *and deposition of H. Kwak, p. 112, 132, 135-137*].

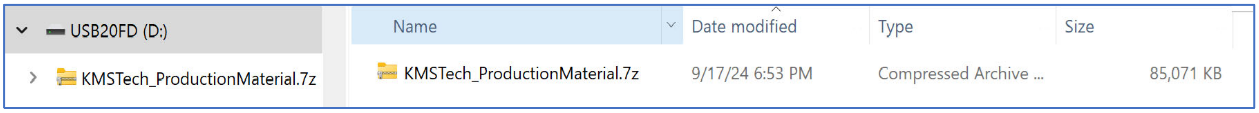
12. Accompanying this report produced are materials relied upon, found in directories as shown below:

Directory within accompanying material	**How obtained**
analysisAndResults	Produced by automated script analysis of this report
fromGMission_Playbox	Retrieved from E: drive of Defendant's Playbox
KmsTech_eFileDeposit	Received from Plaintiff's law firm.



analysisAndResults
fromGMission_PlayBox
KmsTech_eFileDeposit

13. The material has been produced on USB flash drive in a 7Z-archive file as below.



Name	Date modified	Type	Size
KMSTech_ProductionMaterial.7z	9/17/24 6:53 PM	Compressed Archive ...	85,071 KB

To validate the integrity of the received file, the following MD5 checksum can be performed after copying the file from USB and prior to access by using the following Windows DOS "certutil" command:.

```
certutil -hashfile KMSTech_ProductionMaterial.7z MD5
```

14. To verify the integrity of the produced zip-archive file, in a windows command prompt, "CD" to the directory where KMSTech_ProductionMaterial.7z has been copied and invoke the above command. If the file is identical to the material I produced, the MD5 hash will be identical to the hash shown below:

```
MD5 hash of KMSTech_ProductionMaterial.7z:
c8a79292cf463b0b4dd07f8f64dae3c5
CertUtil: -hashfile command completed successfully.
```

15. I reserve the right to rely upon any additional information I become aware of after the date of this report.

V. SUMMARY OF FINDINGS REGARDING CODE OF DEFENDANT'S PLAYBOX

16. The source code obtained by de-compiling the executable d_playbox of the Defendant's system, when compared line-by-line to corresponding elements of the Plaintiff's p_deposit source code is found to be substantially similar [*see infra, section X "LINE-BY-LINE COMPARISON"*].

17. The class method signatures of the source code of d_playbox, when compared to the class method signatures of the p_deposit, are nearly identical. This element of the author's unique creative expression is literally found in the d_playbox and is essentially 100% identical to the corresponding elements of p_deposit [*see, infra section XI, "METHOD SIGNATURE COMPARISON"*].

18. Owing to the substantial similarity, line-by-line (A, above) and the identical method signatures (B, above), it is my opinion that code of the Defendant's d_playbox software, SSMediaPlayer.exe could only have been created using source code copied from the Plaintiff's Copyright Deposit.

19. The d_playbox code files also include an embedded copyright notice [*see, infra VIII "EMBEDDED COPYRIGHT"*].

VI. SUMMARY OF METHOD USED TO PERFORM THE COMPARISONS

20. After reviewing the materials, I decided to perform both a line-by-line comparison of corresponding source code elements [*see infra, X "*LINE-BY-LINE COMPARISON*"*] and also to perform a comparison of the corresponding method signatures [see infra, XI, "METHOD SIGNATURE COMPARISON"]

21. In order to perform the line-by-line comparison, I followed the following steps:

a. I processed program code found on d_playbox (i.e., 'E:\SSMediaPlayer\SSMediaPlayer.exe'), using the Jetbrains decompiler[3]. I refer to the de-compiler output as the d_playbox source code.

b. Using an automated script and manual comparison, I was able to align blocks of source code comprising 96%[4] of the copyright-able synonym matched, d_playbox source code with corresponding code of the p_deposit. The alignment process was performed only once per d_playbox C# source code file, to find which p_deposit file and within that p_deposit file the start and end line that corresponded to the lines

[3] https://web.archive.org/web/20240418164515/https://www.jetbrains.com/decompiler/

[4] See, infra, Figure X-1: Line-by-line compare of d_playbox to copyright p_deposit*, calculated from LOC totals: [M27] / ([M27] + [L28]), i.e. 0.96 = 4561 / (4561 + 217).*

With reference to *Figure X-1* the following categories of files and Lines of Code, ("LOC") were reviewed:

i. Processed by automated script: 19 files of the 25 that were able to be compared using the automated script (*see dPlaybox file enumeration, Figure X-1, Column M and row 27, "...Lines of Code Compared", total 4561*).

a. Within the 19 files of the scripted comparison, i.e. rows 2-9, 12-15, 17-22, there were 217 d_playbox LOC, that when aligned with the pDeposit Copyright source code were excluded from the comparison. Counts for these LOC were noted in column L by the script and totaled in row 28. These lines were the leading, header lines of the files. The differences are because of differences in how the Microsoft C# compiler and the JetBrains decompiler tools processed the source code. These differences were not attributable to developer creative expression.

b. There were 4561 LOC of d_playbox compared line-by-line to p_deposit.

ii. Not processed by script: 6 of the 25 files that did not lend themselves to scripted comparison.

a. Not applicable to copyright: 3 of the 6 not processed by-script files, i.e., rows 10, 11 and 16, LOC, 358 LOC -- totaled in row 29 -- were not applicable to copyright. 2 were automatically generated by the Microsoft C# compiler and 1 was attributable to a well-known Microsoft interface.

b. Manually reviewed: Three of the 6 non-scripted files, i.e., rows 23, 24 and 25, with 72 LOC – totaled in row 30 -- did not lend themselves to easy alignment and comparison line-by-line using the automated script. For these three, I performed a manual, alignment and comparison to obtain column M and N data.

within the d_playbox file. [*See the complete source code file enumeration, infra, Figure X-1: Line-by-line compare of d_playbox to copyright p_deposit.]*

c. The automated script matched commonly used C# synonyms [*see infra, IX "*MATCHING OF C# SYNONYMS*"*].

d. The script used the Python "difflib", one of many variants of DIFF[5], an industry standard utility program for comparison, developed at Bell Labs during the 1970s, to perform a line-by-line, comparison. The line by line source code comparison results are summarized in *Figure X-1: Line-by-line compare of d_playbox to copyright p_deposit,* showing:

- names of d_playbox source files compared
- counts of lines of code compared
- counts of lines of code excluded from this analysis
- percent of lines of code compared that were found to be identical

22. I also performed an exhaustive automated comparison of the hundreds of class method signatures[6] of the d_playbox to the corresponding class method signatures of the p_deposit.

VII. VISIT TO GMISSION

23. On Feb. 12, 2024, at 1PM I visited Defendant's premises accompanied by Eleni Melekou, the Plaintiff's attorney.

24. At Karaoke City (22 West 32nd Street, 7th Floor, NYC 10001) we were met by the Defendant, and his attorney.

[5] The automated comparison script of this analysis used the Python library "difflib" [*see,* *https://docs.python.org/3/library/difflib.html#*], to produce two forms of output, (i) HTML output which can be easily viewed in any browser [*see accompanying production material at "..KMSTech_ProductionMaterial/analysisAndResults/html/*"*] and (ii) "unified" output [*see accompanying production material at "..KMSTech_ProductionMaterial/analysisAndResults/unified/*"*]. The script processed the "unified" output in order to calculate the count summary data [*see accompanying production material at "..KMSTech_ProductionMaterial/analysisAndResults/loc_counts.csv and an annotated version of that same analysis result data found infra, Figure X-1: Line-by-line compare of d_playbox to copyright p_deposit, p.23*].

For history and additional information regarding the DIFF algorithms and variants, see:

- https://web.archive.org/web/20240909172842/https://thelinuxcode.com/linux-diff-command-examples/
- https://web.archive.org/web/20191225170546/https://www.gnu.org/software/diffutils/manual/diffutils.pdf

[6] The method signatures exhibit the developer's creative expression in a concise form, as each signature includes a minimum of 5 distinct and unique choices made by the author of the source code. (*See infra section XI.*)

25. The Defendant began starting up a system in a small closet-sized room with multiple computers, multiple display monitors and other electronic equipment.

26. I asked the Defendant's attorney if the system I was being provided to review contained the Defendant's "copy" of the Plaintiff's system as I understood "copy" from reviewing relevant parts of the transcript of Mr. Kwak's (the Defendant's) deposition[7]. The Defendant's attorney went to another room and when he returned, he responded that the system they were providing for my review was the system that had been provided to the Defendant by the Plaintiff, i.e. d_playbox.

27. Figure VII-1 thru Figure VII-3, are images of the system produced during the visit.

28. This system was labeled on the topside as "Playbox Karaoke Rental", and "ROOM01" (Figure VII-1, below)

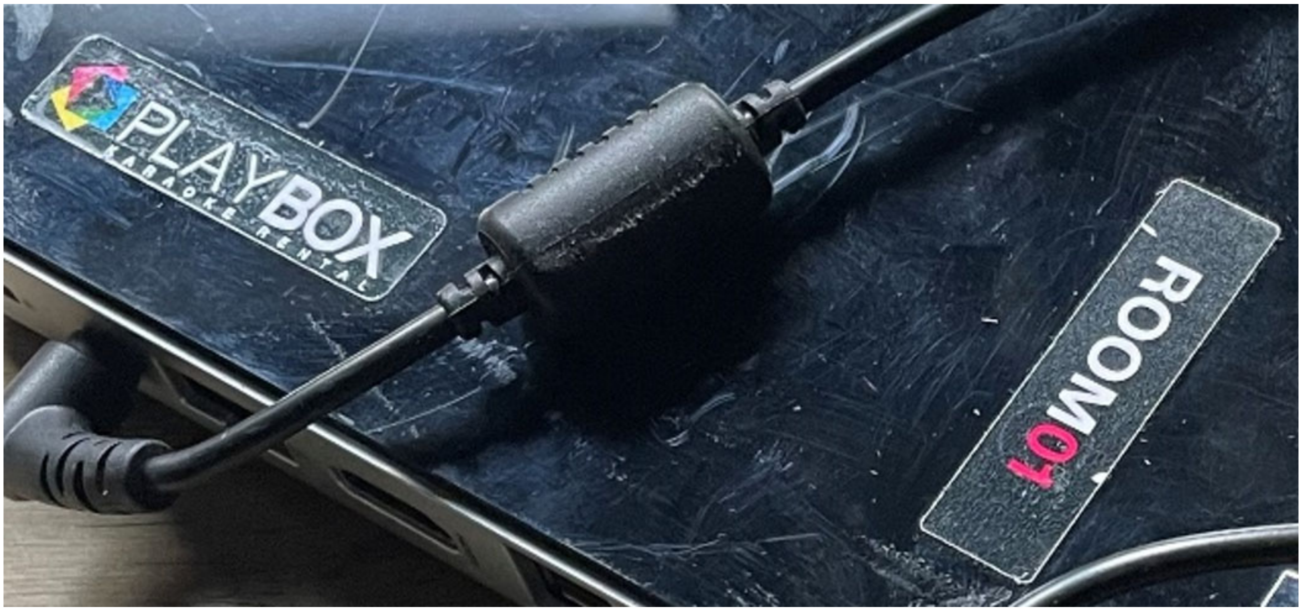


Figure VII-1: Plaintiff premise system labels

[7] See, transcript content relevant to "copying", for example, *Deposition of H. Kwak*, pp. 112, 132, 135-137

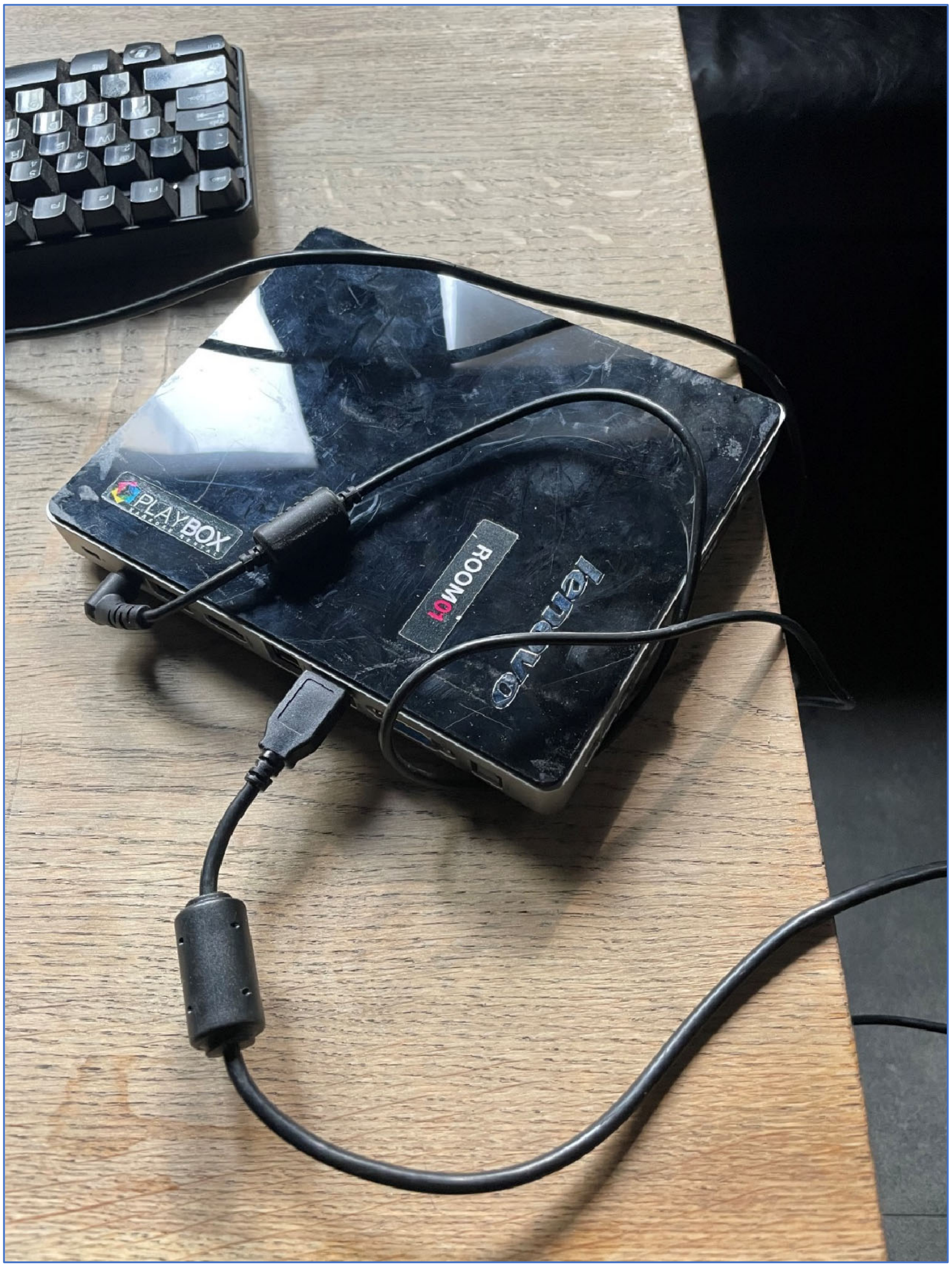


Figure VII-2: Topside "PLAYBOX KARAOKE RENTAL" at Plaintiff's premises

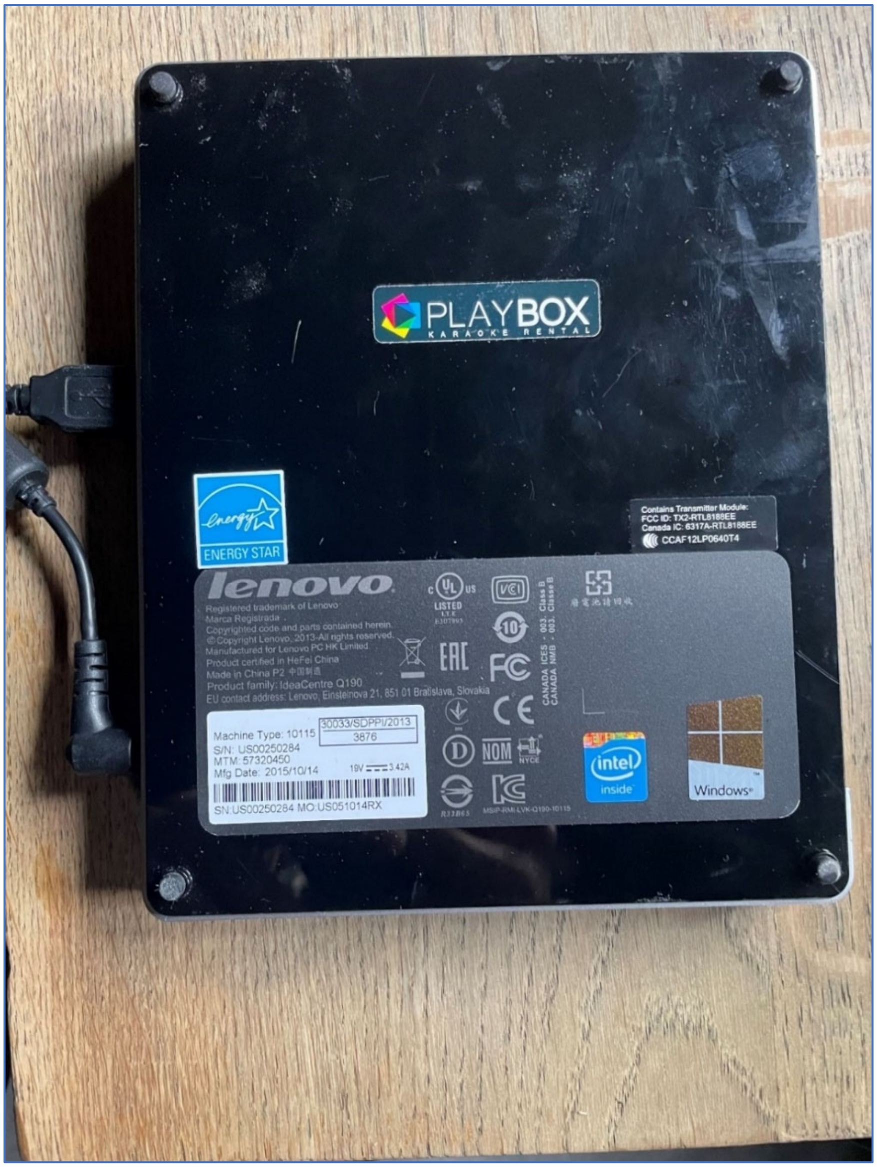

PLAYBOX
KARAOKE RENTAL
ENERGY STAR
Contains Transmitter Module:
FCC ID: TX2-RTL8188EE
Canada IC: 6317A-RTL8188EE
CCAF12LP0640T4
lenovo
Registered trademark of Lenovo
Marca Registrada
Copyrighted code and parts contained herein.
© Copyright Lenovo, 2013-All rights reserved.
Manufactured for Lenovo PC HK Limited
Product certified in HeFei China
Made in China P2 中国制造
Product family: IdeaCentre Q190
EU contact address: Lenovo, Einsteinova 21, 851 01 Bratislava, Slovakia
Machine Type: 10115
30033/SDPPI/2013
3876
S/N: US00250284
MTM: 57320450
Mfg Date: 2015/10/14
19V 3.42A
SN:US00250284 MO:US051014RX
LISTED
CANADA ICES - 003, Class B
CANADA NMB - 003, Classe B
NOM
NYCE
R33B65
MSIP-RMI-LVK-Q190-10115
intel
inside
Windows

Figure VII-3: Underside "PLAYBOX KARAOKE RENTAL" at Plaintiff premises

29. I was able[8] to extract files from d_playbox, onto a USB thumb drive. One of those files retrieved was named "SSMediaPlayer.exe and the contents of the containing E: drive directory named "SSMediaPlayer"[9].

[8] The system was configured to impede direct access to the file system. The Windows start up entered a "kiosk" mode in which user access to the underlying system was not available. By using a system boot interrupt keystroke within a 3-second start up window, I was able to get access to the file system.

[9] See accompanying production material at the path, "..\KMSTech_ProductionMaterial\fromGMission_PlayBox\SSMediaPlayer\"

30. At the completion of the extraction, I *again* asked if I could review the Defendant systems that were in each of the Defendant Restaurant’s Karaoke rooms or review the systems with the “copy”. I was informed by the Defendant’s attorney that would not occur.

31. The limitation by the Defendant’s attorney of my visit to review only the D_Playbox system and not also the other systems at the Defendant’s premises, as noted in the relevant court order, [*see ECF 158*], precluded determination of answers to questions such as:

a. How were the copies used?
b. Where were the copies used?
c. How had the copies been obtained from the D_Playbox system?
d. How many copies had been made?
e. How were those copies packaged and installed?
f. Was it easy for those copies to then be provided to third parties?

32. Any literal “copy” of the D_Playbox software[10]-- if subjected to the analysis of this report, would result in the same conclusions as from my analysis of D_Playbox, i.e. those literal copies were made using source code copied from Plaintiff's Copyright Deposit.

33. Failure to permit my review of the other systems at the Defendant's premises prevents me from making a determination that as of the date of my visit, those systems DID NOT use copies of the Plaintiff's Copyright Deposit or somehow were not substantially similar.

34. If the Defendant's defense is that their Karaoke system includes elements that were not literally copied, and that somehow the portions copied are not substantially similar to P_Deposit, then that hypothetical Defendant system would need to be provided to me for review. However, I asked twice for any such systems to review during my visit and my requests were not honored.

VIII. EMBEDDED COPYRIGHT

35. Upon examination of the SSMediaPlayer.exe file obtained from d_playbox system a copyright is easily viewed using the native Microsoft system file explorer, without any additional tools. This can be accomplished with the following steps and as shown in the following screen shots:

[10] For example, literal copies such as, (i) a literal copy of the file ‘ssmediaplayer.exe’ software found on D_Playbox or (ii) a literal cloning of the entire D_Playbox system or (iii) even a run-time reference from the Defendant’s systems to the D_Playbox system operating as a file-server thereby repeatedly performing literal copying with each access, e.g. via a filesystem, network or the Internet.

- Using the Microsoft built-in file explorer, select (single left-click) on the file 'SSMediaPlayer.exe' (*see Figure VIII-1, up pointing red arrow annotation*)
- "right-click" when the SSMediaPlayer.exe file is highlighted in grey. This displays a pop-up where one choice is "**Properties**" (*see Figure VIII-1, right pointing red arrow annotation*)
- Selecting "**Properties"** drop-down choice will display a pop-up dialog display, with the "**General**" tab selected by default (*see Figure VIII-2).*

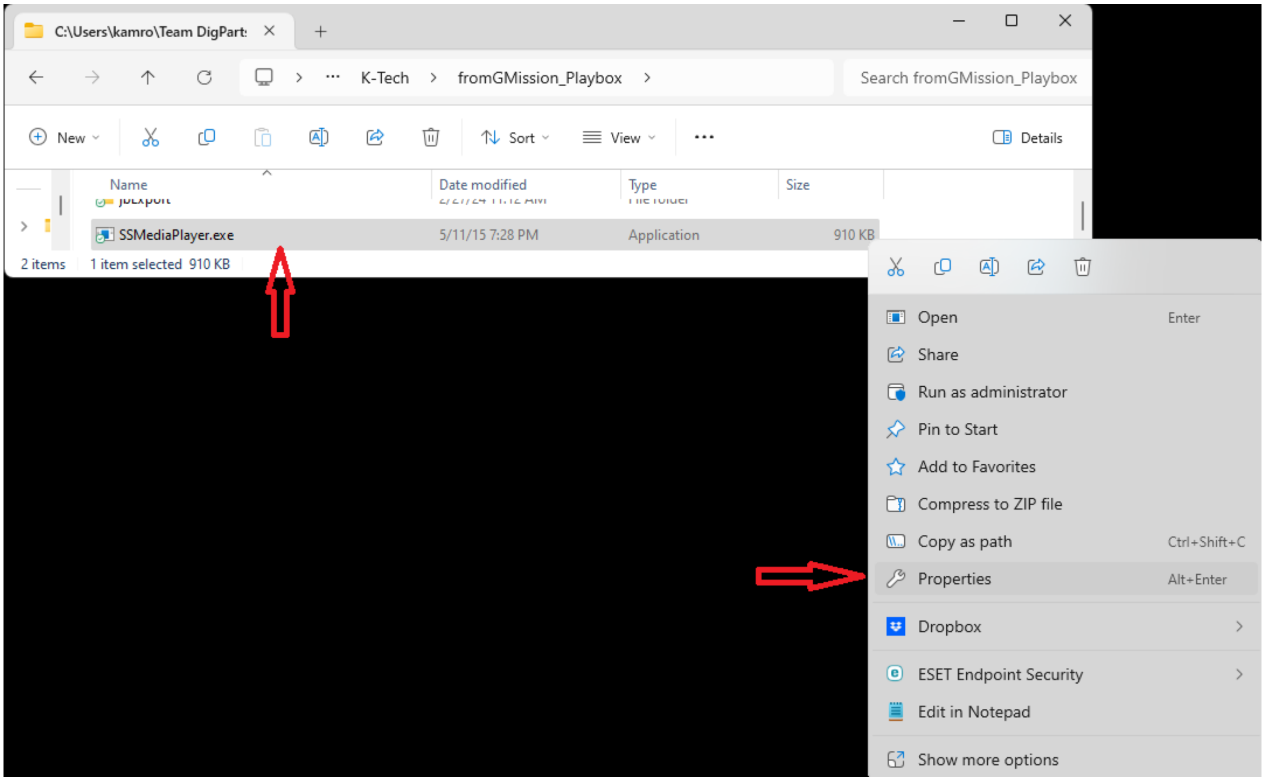
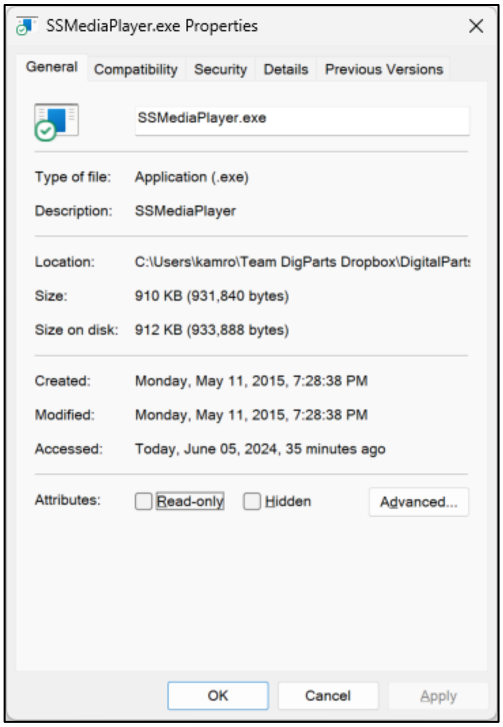


Figure VIII-1. Using File Explorer to view "Properties" display dialogue.



SSMediaPlayer.exe

Type of file: Application (.exe)

Description: SSMediaPlayer

Location: C:\Users\kamro\Team DigParts Dropbox\DigitalParts

Size: 910 KB (931,840 bytes)

Size on disk: 912 KB (933,888 bytes)

Created: Monday, May 11, 2015, 7:28:38 PM

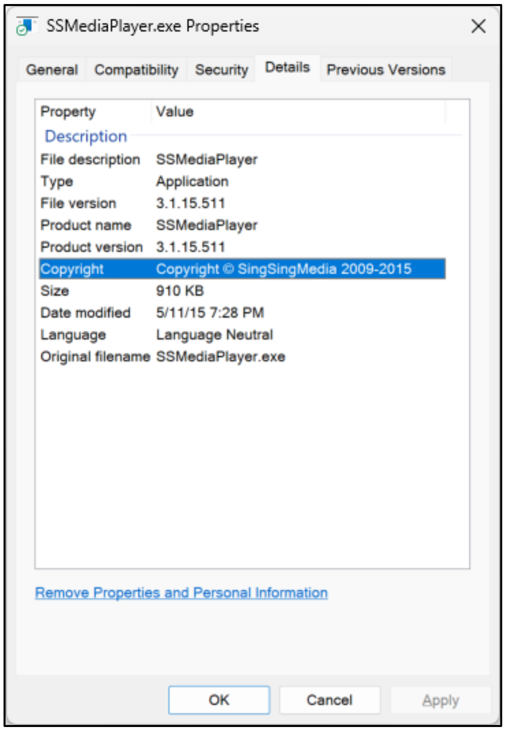
Modified: Monday, May 11, 2015, 7:28:38 PM

Accessed: Today, June 05, 2024, 35 minutes ago

Attributes: ☐ Read-only ☐ Hidden

Figure VIII-2. Windows file Windows "General" Properties for SSMediaPlayer.exe

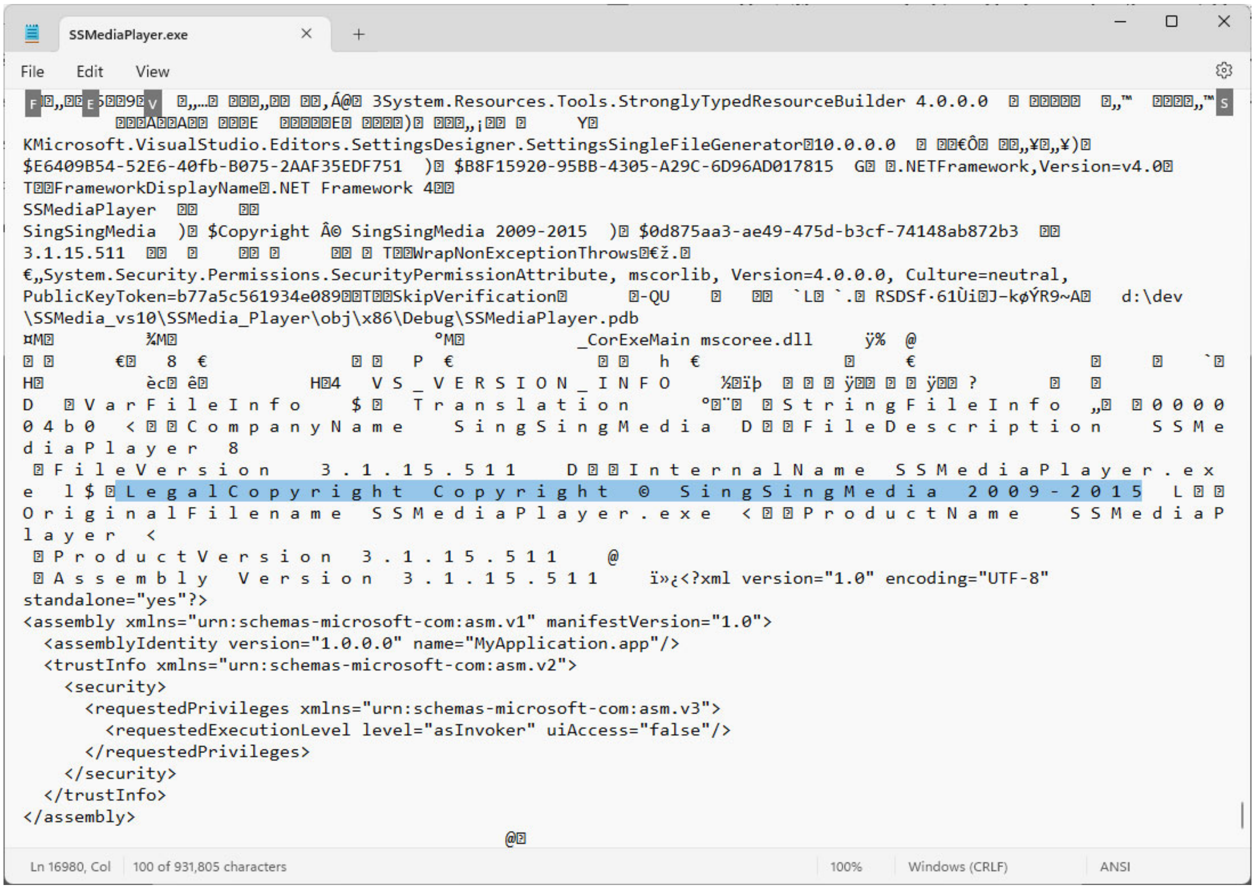
- Selecting the "**Details**" tab, shown above, displays a pop-up dialog that shows the embedded copyright along with other properties (*see below, Figure VIII-3*).


SSMediaPlayer.exe Properties
General
Compatibility
Security
Details
Previous Versions
Remove Properties and Personal Information
OK
Cancel
Apply

Property	Value
Description	
File description	SSMediaPlayer
Type	Application
File version	3.1.15.511
Product name	SSMediaPlayer
Product version	3.1.15.511
Copyright	Copyright © SingSingMedia 2009-2015
Size	910 KB
Date modified	5/11/15 7:28 PM
Language	Language Neutral
Original filename	SSMediaPlayer.exe

Figure VIII-3. Windows file system "Details" Properties for SSMediaPlayer.exe

36. An alternate way to view the copyright is to simply open the file ‘SSMediaPlayer.exe’ using the system Notepad editor. The last few lines of the file content visible in the editor (*see Figure VIII-4*) exhibit this copyright.



```
F□„□□E5□□9□V  □„…□ □□□„□□ □□,Á@□ 3System.Resources.Tools.StronglyTypedResourceBuilder 4.0.0.0  □ □□□□□  □„™  □□□□„™S
         □□□A□□A□□ □□□E  □□□□□E□ □□□□)□ □□□„¡□□ □      Y□
KMicrosoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator□10.0.0.0  □ □□€Ô□ □□„¥□„¥)□
$E6409B54-52E6-40fb-B075-2AAF35EDF751  )□ $B8F15920-95BB-4305-A29C-6D96AD017815  G□ □.NETFramework,Version=v4.0□
T□□FrameworkDisplayName□.NET Framework 4□□
SSMediaPlayer  □□   □□
SingSingMedia  )□ $Copyright Â© SingSingMedia 2009-2015  )□ $0d875aa3-ae49-475d-b3cf-74148ab872b3  □□
3.1.15.511  □□  □    □□ □     □□ □ T□□WrapNonExceptionThrows□€ž.□
€„System.Security.Permissions.SecurityPermissionAttribute, mscorlib, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=b77a5c561934e089□□T□□SkipVerification□     □-QU   □   □□  `L□ `.□ RSDSf·61Ùi□J-køÝR9~A□   d:\dev
\SSMedia_vs10\SSMedia_Player\obj\x86\Debug\SSMediaPlayer.pdb
¤M□       ¾M□                    °M□              _CorExeMain mscoree.dll     ÿ%  @
□ □     €□   8  €             □ □   P  €             □ □   h  €             □     €              □     □    `□
H□        èc□ ê□         H□4   V S _ V E R S I O N _ I N F O     ½□ïþ  □ □ □ ÿ□□ □ □ ÿ□□ ?      □   □
D   □ V a r F i l e I n f o     $ □   T r a n s l a t i o n      °□¨□  □ S t r i n g F i l e I n f o   „□  □ 0 0 0 0
0 4 b 0   < □ □ C o m p a n y N a m e     S i n g S i n g M e d i a   D □ □ F i l e D e s c r i p t i o n     S S M e
d i a P l a y e r   8
 □ F i l e V e r s i o n     3 . 1 . 1 5 . 5 1 1     D □ □ I n t e r n a l N a m e   S S M e d i a P l a y e r . e x
e   1 $ □ L e g a l C o p y r i g h t   C o p y r i g h t   ©   S i n g S i n g M e d i a   2 0 0 9 - 2 0 1 5   L □ □
O r i g i n a l F i l e n a m e   S S M e d i a P l a y e r . e x e   < □ □ P r o d u c t N a m e     S S M e d i a P
l a y e r   <
 □ P r o d u c t V e r s i o n   3 . 1 . 1 5 . 5 1 1    @
 □ A s s e m b l y   V e r s i o n   3 . 1 . 1 5 . 5 1 1     ï»¿<?xml version="1.0" encoding="UTF-8"
standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <assemblyIdentity version="1.0.0.0" name="MyApplication.app"/>
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
    <security>
      <requestedPrivileges xmlns="urn:schemas-microsoft-com:asm.v3">
        <requestedExecutionLevel level="asInvoker" uiAccess="false"/>
      </requestedPrivileges>
    </security>
  </trustInfo>
</assembly>
                                                   @□
```

Figure VIII-4. Copyright visible for file, ‘SSMediaPlayer.exe’ using any text editor.

IX. MATCHING OF C# SYNONYMS

37. Use of synonyms in C# is akin to paraphrasing in natural language. English language paraphrase examples are:

Original Sentence: **“**James Smith went to the concert to hear music**”**
Paraphrased: “Jim went to the concert to hear some music.”

Original Sentence: “The scientific community reached a consensus on climate change.”
Paraphrased: “Scientists all agreed about climate change.”

Original Sentence: “The sun rises in the east and sets in the west.”
Paraphrased: “The sun comes up over there and goes down over here.”

38. One category of C# language synonyms, is differently specified object hierarchies. The JetBrains decompiler uses fully specified object hierarchies, whereas the Microsoft compiler allows less than fully specified hierarchies. The source code supplies the full specification resolving these less than fully specified symbols with reference to its context in the source code.

39. In this category of symbol / scope hierarchy are examples such as:

- “System.Windows.Forms.”
- “Windows.Forms.”
- “Forms.”

40. These can be identical expressions for the C# object “System.Windows.Forms.

41. In C# language, another synonym category is the use of variable number of spaces, tabs or blank lines. The volume of white space, beyond a single space, is all that same as a single space and variations are not a creative expression of the author.

42. A limited set of synonym matching was performed by the automated script that included the following categories:

a. Matching of white space, such as blank lines, spaces or tabs.

b. Removal of comments

c. Object scoping matches:

i. Matching of local “this” scoping, e.g.

- “this.X”

- “X”

ii. Matching self-class, implicit scoping, for example, within a class named “Program” e.g., the synonyms:

- “Program.logger”
- “logger” within the class named “Program”

iii. Matching of explicit scoping for .NET system and other library objects, e.g. ¶ 39 above.

d. Matching explicit variable type casts, for the same numerical value, e.g.

- “(byte) 0”
- “0”

e. Matching single-line lamda function definitions, to multi-line definitions, e.g. note yellow highlighted differences in these identical, (matched) expressions

```
•     private void btnExit_Click_1(object sender, EventArgs
e) => logger.Debug("Exit button clicked");
•     private void btnExit_Click_1(object sender, EventArgs
e)
     {
     logger.Debug("Exit button clicked”);
     }
```

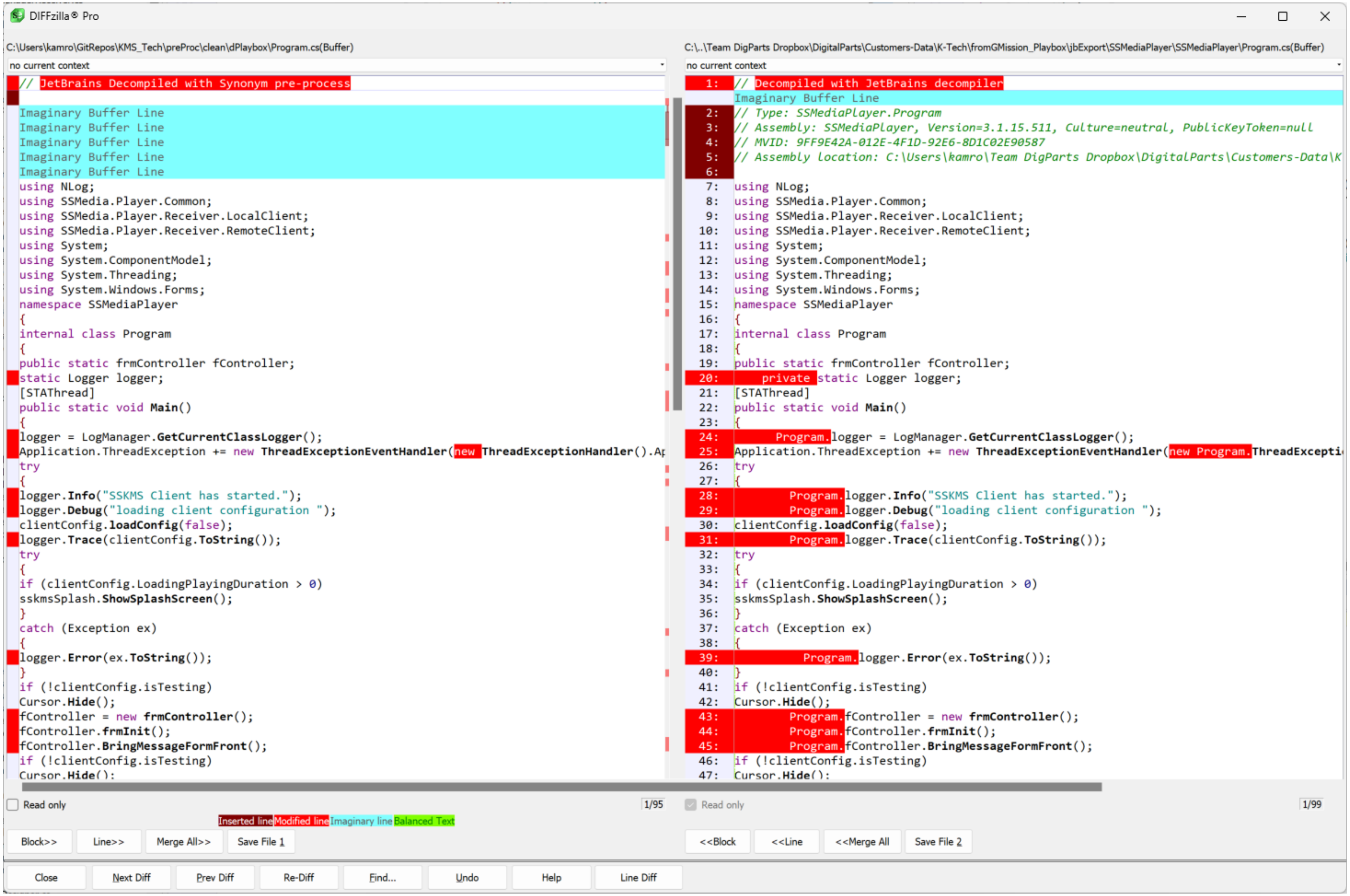
43. In the screenshot comparisons[11], (*Figure IX-1 and Figure IX-2, below)* the red indicates revisions made to match synonyms, whereas lines without any red are identical.[12].

44. Figure IX-1, below, exhibits the raw **d_playbox** source code (*right side of screenshot*) compared to its synonym matched version (left side of screenshot) for the “Program” class.

45. Figure IX-2, below, exhibits the raw **p_deposit** source code (right side of screenshot) compared to its synonym matched version (left side of screenshot) for the “Program” class.

[11] These screenshots, (*i.e. Figure IX-1: Example annotation of synonym matching of d_playbox* source code*, infra p. 19 and Figure IX-2: Example annotation of synonym matching of p_deposit* source code*, infra p. 20*), use a text editor tool DIFF algorithm variant [*see supra footnote 5*] to perform the comparison. The name of this variant is “SlickEdit Diffzilla Pro [*see, https://web.archive.org/web/20240716230510/https://www.slickedit.com/compare-editions#:~:text=(Pro%20only)%20View%20and%20reconcile,e.g.%20function%20definitions)%20to%20diff.*]

[12] The comparison screenshot examples of this section highlight the matching of C# synonyms performed prior to the line-by-line comparison of d_playbox to p_deposit. The filenames of these examples can be seen near the top of the screenshots.



C:\Users\kamro\GitRepos\KMS_Tech\preProc\clean\dPlaybox\Program.cs(Buffer)

```
// JetBrains Decompiled with Synonym pre-process

Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
using NLog;
using SSMedia.Player.Common;
using SSMedia.Player.Receiver.LocalClient;
using SSMedia.Player.Receiver.RemoteClient;
using System;
using System.ComponentModel;
using System.Threading;
using System.Windows.Forms;
namespace SSMediaPlayer
{
internal class Program
{
public static frmController fController;
static Logger logger;
[STAThread]
public static void Main()
{
logger = LogManager.GetCurrentClassLogger();
Application.ThreadException += new ThreadExceptionEventHandler(new ThreadExceptionHandler().Ap
try
{
logger.Info("SSKMS Client has started.");
logger.Debug("loading client configuration ");
clientConfig.loadConfig(false);
logger.Trace(clientConfig.ToString());
try
{
if (clientConfig.LoadingPlayingDuration > 0)
sskmsSplash.ShowSplashScreen();
}
catch (Exception ex)
{
logger.Error(ex.ToString());
}
if (!clientConfig.isTesting)
Cursor.Hide();
fController = new frmController();
fController.frmInit();
fController.BringMessageFormFront();
if (!clientConfig.isTesting)
Cursor.Hide():
```

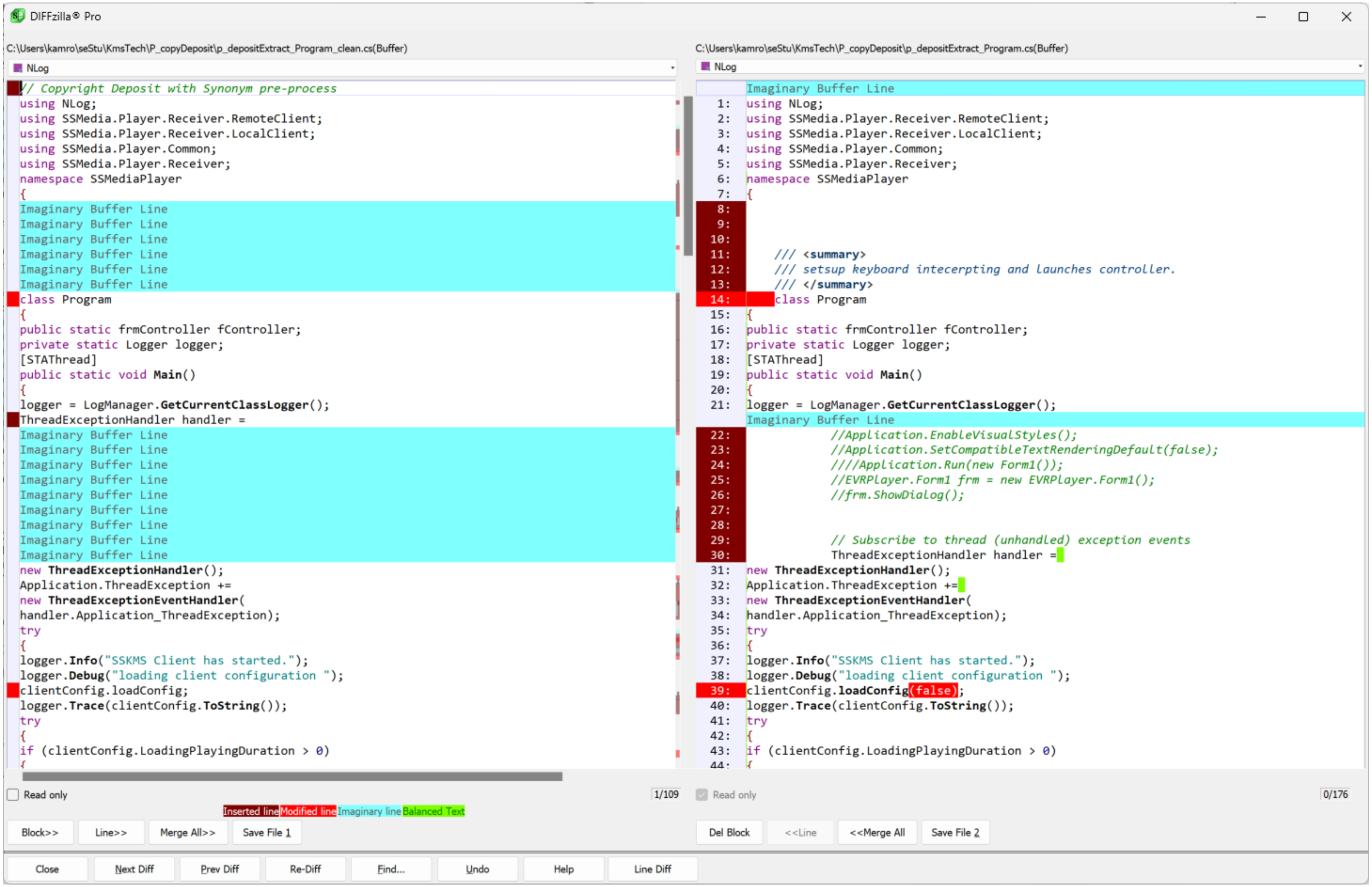
C:\..\Team DigParts Dropbox\DigitalParts\Customers-Data\K-Tech\fromGMission_Playbox\jbExport\SSMediaPlayer\SSMediaPlayer\Program.cs(Buffer)

```
// Decompiled with JetBrains decompiler
Imaginary Buffer Line
// Type: SSMediaPlayer.Program
// Assembly: SSMediaPlayer, Version=3.1.15.511, Culture=neutral, PublicKeyToken=null
// MVID: 9FF9E42A-012E-4F1D-92E6-8D1C02E90587
// Assembly location: C:\Users\kamro\Team DigParts Dropbox\DigitalParts\Customers-Data\K

using NLog;
using SSMedia.Player.Common;
using SSMedia.Player.Receiver.LocalClient;
using SSMedia.Player.Receiver.RemoteClient;
using System;
using System.ComponentModel;
using System.Threading;
using System.Windows.Forms;
namespace SSMediaPlayer
{
internal class Program
{
public static frmController fController;
    private static Logger logger;
[STAThread]
public static void Main()
{
      Program.logger = LogManager.GetCurrentClassLogger();
Application.ThreadException += new ThreadExceptionEventHandler(new Program.ThreadExcepti
try
{
        Program.logger.Info("SSKMS Client has started.");
        Program.logger.Debug("loading client configuration ");
clientConfig.loadConfig(false);
        Program.logger.Trace(clientConfig.ToString());
try
{
if (clientConfig.LoadingPlayingDuration > 0)
sskmsSplash.ShowSplashScreen();
}
catch (Exception ex)
{
          Program.logger.Error(ex.ToString());
}
if (!clientConfig.isTesting)
Cursor.Hide();
        Program.fController = new frmController();
        Program.fController.frmInit();
        Program.fController.BringMessageFormFront();
if (!clientConfig.isTesting)
Cursor.Hide():
```

Figure IX-1: Example annotation of synonym matching of d_playbox source code

(Left, after matching; Right is prior to matching.)



C:\Users\kamro\seStu\KmsTech\P_copyDeposit\p_depositExtract_Program_clean.cs(Buffer)

```
// Copyright Deposit with Synonym pre-process
using NLog;
using SSMedia.Player.Receiver.RemoteClient;
using SSMedia.Player.Receiver.LocalClient;
using SSMedia.Player.Common;
using SSMedia.Player.Receiver;
namespace SSMediaPlayer
{
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
class Program
{
public static frmController fController;
private static Logger logger;
[STAThread]
public static void Main()
{
logger = LogManager.GetCurrentClassLogger();
ThreadExceptionHandler handler =
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
Imaginary Buffer Line
new ThreadExceptionHandler();
Application.ThreadException +=
new ThreadExceptionEventHandler(
handler.Application_ThreadException);
try
{
logger.Info("SSKMS Client has started.");
logger.Debug("loading client configuration ");
clientConfig.loadConfig;
logger.Trace(clientConfig.ToString());
try
{
if (clientConfig.LoadingPlayingDuration > 0)
{
```

C:\Users\kamro\seStu\KmsTech\P_copyDeposit\p_depositExtract_Program.cs(Buffer)

```
Imaginary Buffer Line
using NLog;
using SSMedia.Player.Receiver.RemoteClient;
using SSMedia.Player.Receiver.LocalClient;
using SSMedia.Player.Common;
using SSMedia.Player.Receiver;
namespace SSMediaPlayer
{



    /// <summary>
    /// setsup keyboard intecerpting and launches controller.
    /// </summary>
    class Program
{
public static frmController fController;
private static Logger logger;
[STAThread]
public static void Main()
{
logger = LogManager.GetCurrentClassLogger();
Imaginary Buffer Line
            //Application.EnableVisualStyles();
            //Application.SetCompatibleTextRenderingDefault(false);
            ////Application.Run(new Form1());
            //EVRPlayer.Form1 frm = new EVRPlayer.Form1();
            //frm.ShowDialog();


            // Subscribe to thread (unhandled) exception events
            ThreadExceptionHandler handler =
new ThreadExceptionHandler();
Application.ThreadException +=
new ThreadExceptionEventHandler(
handler.Application_ThreadException);
try
{
logger.Info("SSKMS Client has started.");
logger.Debug("loading client configuration ");
clientConfig.loadConfig(false);
logger.Trace(clientConfig.ToString());
try
{
if (clientConfig.LoadingPlayingDuration > 0)
{
```

Figure IX-2: Example annotation of synonym matching of p_deposit source code

(Left, after matching; Right is prior to matching.)

X. LINE-BY-LINE COMPARISON

46. The automated script first corresponded the start and end line numbers of similar source code elements (e.g. lines of code of identically named classes) of d_playbox and p_deposit. The python script difflib tool (*see, supra, footnote 5)* was then used to perform the comparison of d_playbox source code to p_deposit. For each d_playbox class file, difflib compared the d_playbox lines, to the corresponding lines found in p_deposit files.

47. The line-counts and results of these line-by-line comparisons are found in Figure X-1, on p. 24, below.

48. The columns of that table have the following meaning:

Col	Caption	Meaning
A	dPlaybox file compared	Name of file from dPlaybox compared
B	sort Seq	Sequence of dPlaybox file processed.
C	File "LOC" Lines of Code	Count of Lines of Code of the file
D	File Type	Type of source code found in the file. "**class**" files are one programmatic element of the C# language. A C# class is a container of creative expression of the program's author. It exhibits the author's original creativity in his/her choice of symbolic names, the name of the class, the name of its variables, the method names and the method-signatures and even the sequence of the methods and variable declarations. These class files were recognized by the automated scripted alignment and comparison and the results of that comparison were written to several columns of the table of Figure X-1, specifically, columns C, G, H, I, J, K, O. "**names**" files contained symbolic definitions. Both of these files contained compiler generated code, not authored by a developer. These lines of code could not be expected to be creatively unique and so aside from manually reviewing them and counting the lines of code, I excluded them from my consideration. "**wrap**" contained methods that were a "wrapper" of the well-known Micorosoft DirectShow interface. This well-known interface could not be expected to be creatively unique and so aside from manually reviewing it and counting the lines of code, I excluded these lines of code from my consideration. "**iface**" and "**enum**" contained symbolic definitions and interfaces. I had to compare the content of these files manually in order to align them with the appropriate start and end lines of the Copyright source code files, as they were not recognized by the automated script. For these files, I then manually compared line-by-line and produced the data of the corresponding columns of Figure X-1.
E	Compare method	"Scripted" were processed automatically as described herein, using difflib. "NA" were excluded from the comparison results. "Manual" were reviewed manually.
F	className	If the filetype (column E) is "class", then this is the name of the class used for the automated alignment. Else this is blank
G	dPlaybox StartLine	StartLine of file named in (A) for line-by-line comparison
H	dPlaybox EndLine	EndLine of file named in (A) for line-by-line comparison
I	pDeposit filename	Name of pDeposit file that contains corresponding lines of code found in the Copyright deposit.

J	pDeposit StartLine	StartLine of file named (I) for line-by-line comparison
K	pDeposit EndLine	EndLine of file named (I) for line-by-line comparison
L	dPlaybox LOC NOT Compared	This is the count of any LOC that may have been skipped of dPlaybox file
M	dPlaybox LOC Compared	This is the count of LOC compared line-by-line of dPlaybox file
N	Line-by-line % D identical	Percent of LOC compared from dPlaybox that were identical (and in the same sequence) to LOC in pDeposit
O	LOC Identical	Total LOC compared from dPlaybox that were identical (and in the same sequence) to LOC in PDeposit

A. Identical Lines of Code

49. The aggregate result for identical lines of code of dPlaybox compared to pDeposit is found in Figure X-1, row 27.

50. 3269 lines of code out of 4561 lines of code compared were identical, that is 72% of the lines of code of d_playbox were found to be identical to lines of code of p_deposit, after alignment and synonym matching.

51. See footnote 4, above on p. 9 for details regarding the lines of code reported in that table.

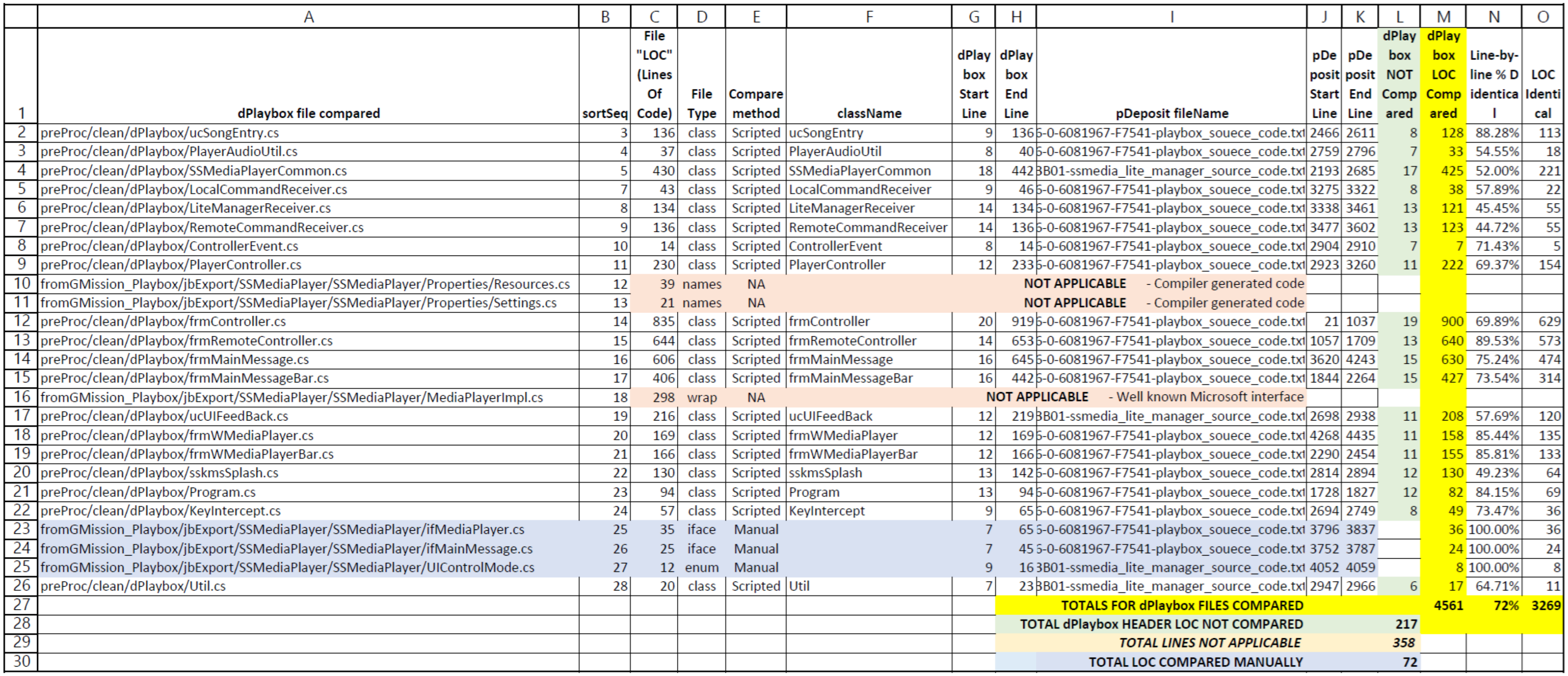
52. The reported "identical" LOC is a lower-bound metric for similarity. Because of many synonyms not detected by the automated script, (*see* ¶¶ *53, 54 and 55*), the scripted, overall, 72% LOC "identical" is a conservative estimate of similarity.

B. Unmatched Synonyms - Lines of Code Not Identical

53. Many lines that do not match identically after the limited pre-processing synonym matching, were spot checked and found to be synonyms that, owing to their syntactic complexity were unmatched by the automated script. These unmatched synonyms are related to differences of the Microsoft C# compiler and the JetBrains C# decompiler tools that processed the code.

54. Some lines of code that do not match were found to be trivial revisions of p_deposit, some of those also unmatched synonyms, and overall these revisions do not impact the creative expression of p_deposit.

55. The scripted line-by-line comparison lines identified as "not identical", but which are in-fact unmatched synonyms contribute to the calculated 28% LOC "not identical" result.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	dPlaybox file compared	sortSeq	File "LOC" (Lines Of Code)	File Type	Compare method	className	dPlay box Start Line	dPlay box End Line	pDeposit fileName	pDeposit Start Line	pDeposit End Line	dPlay box NOT Compared	dPlay box LOC Compared	Line-by-line % D identical	LOC Identical
2	preProc/clean/dPlaybox/ucSongEntry.cs	3	136	class	Scripted	ucSongEntry	9	136	5-0-6081967-F7541-playbox_souece_code.txt	2466	2611	8	128	88.28%	113
3	preProc/clean/dPlaybox/PlayerAudioUtil.cs	4	37	class	Scripted	PlayerAudioUtil	8	40	5-0-6081967-F7541-playbox_souece_code.txt	2759	2796	7	33	54.55%	18
4	preProc/clean/dPlaybox/SSMediaPlayerCommon.cs	5	430	class	Scripted	SSMediaPlayerCommon	18	442	3B01-ssmedia_lite_manager_source_code.txt	2193	2685	17	425	52.00%	221
5	preProc/clean/dPlaybox/LocalCommandReceiver.cs	7	43	class	Scripted	LocalCommandReceiver	9	46	5-0-6081967-F7541-playbox_souece_code.txt	3275	3322	8	38	57.89%	22
6	preProc/clean/dPlaybox/LiteManagerReceiver.cs	8	134	class	Scripted	LiteManagerReceiver	14	134	5-0-6081967-F7541-playbox_souece_code.txt	3338	3461	13	121	45.45%	55
7	preProc/clean/dPlaybox/RemoteCommandReceiver.cs	9	136	class	Scripted	RemoteCommandReceiver	14	136	5-0-6081967-F7541-playbox_souece_code.txt	3477	3602	13	123	44.72%	55
8	preProc/clean/dPlaybox/ControllerEvent.cs	10	14	class	Scripted	ControllerEvent	8	14	5-0-6081967-F7541-playbox_souece_code.txt	2904	2910	7	7	71.43%	5
9	preProc/clean/dPlaybox/PlayerController.cs	11	230	class	Scripted	PlayerController	12	233	5-0-6081967-F7541-playbox_souece_code.txt	2923	3260	11	222	69.37%	154
10	fromGMission_Playbox/jbExport/SSMediaPlayer/SSMediaPlayer/Properties/Resources.cs	12	39	names	NA				**NOT APPLICABLE** - Compiler generated code						
11	fromGMission_Playbox/jbExport/SSMediaPlayer/SSMediaPlayer/Properties/Settings.cs	13	21	names	NA				**NOT APPLICABLE** - Compiler generated code						
12	preProc/clean/dPlaybox/frmController.cs	14	835	class	Scripted	frmController	20	919	5-0-6081967-F7541-playbox_souece_code.txt	21	1037	19	900	69.89%	629
13	preProc/clean/dPlaybox/frmRemoteController.cs	15	644	class	Scripted	frmRemoteController	14	653	5-0-6081967-F7541-playbox_souece_code.txt	1057	1709	13	640	89.53%	573
14	preProc/clean/dPlaybox/frmMainMessage.cs	16	606	class	Scripted	frmMainMessage	16	645	5-0-6081967-F7541-playbox_souece_code.txt	3620	4243	15	630	75.24%	474
15	preProc/clean/dPlaybox/frmMainMessageBar.cs	17	406	class	Scripted	frmMainMessageBar	16	442	5-0-6081967-F7541-playbox_souece_code.txt	1844	2264	15	427	73.54%	314
16	fromGMission_Playbox/jbExport/SSMediaPlayer/SSMediaPlayer/MediaPlayerImpl.cs	18	298	wrap	NA				**NOT APPLICABLE** - Well known Microsoft interface						
17	preProc/clean/dPlaybox/ucUIFeedBack.cs	19	216	class	Scripted	ucUIFeedBack	12	219	3B01-ssmedia_lite_manager_source_code.txt	2698	2938	11	208	57.69%	120
18	preProc/clean/dPlaybox/frmWMediaPlayer.cs	20	169	class	Scripted	frmWMediaPlayer	12	169	5-0-6081967-F7541-playbox_souece_code.txt	4268	4435	11	158	85.44%	135
19	preProc/clean/dPlaybox/frmWMediaPlayerBar.cs	21	166	class	Scripted	frmWMediaPlayerBar	12	166	5-0-6081967-F7541-playbox_souece_code.txt	2290	2454	11	155	85.81%	133
20	preProc/clean/dPlaybox/sskmsSplash.cs	22	130	class	Scripted	sskmsSplash	13	142	5-0-6081967-F7541-playbox_souece_code.txt	2814	2894	12	130	49.23%	64
21	preProc/clean/dPlaybox/Program.cs	23	94	class	Scripted	Program	13	94	5-0-6081967-F7541-playbox_souece_code.txt	1728	1827	12	82	84.15%	69
22	preProc/clean/dPlaybox/KeyIntercept.cs	24	57	class	Scripted	KeyIntercept	9	65	5-0-6081967-F7541-playbox_souece_code.txt	2694	2749	8	49	73.47%	36
23	fromGMission_Playbox/jbExport/SSMediaPlayer/SSMediaPlayer/ifMediaPlayer.cs	25	35	iface	Manual		7	65	5-0-6081967-F7541-playbox_souece_code.txt	3796	3837		36	100.00%	36
24	fromGMission_Playbox/jbExport/SSMediaPlayer/SSMediaPlayer/ifMainMessage.cs	26	25	iface	Manual		7	45	5-0-6081967-F7541-playbox_souece_code.txt	3752	3787		24	100.00%	24
25	fromGMission_Playbox/jbExport/SSMediaPlayer/SSMediaPlayer/UIControlMode.cs	27	12	enum	Manual		9	16	3B01-ssmedia_lite_manager_source_code.txt	4052	4059		8	100.00%	8
26	preProc/clean/dPlaybox/Util.cs	28	20	class	Scripted	Util	7	23	3B01-ssmedia_lite_manager_source_code.txt	2947	2966	6	17	64.71%	11
27									**TOTALS FOR dPlaybox FILES COMPARED**				**4561**	**72%**	**3269**
28									**TOTAL dPlaybox HEADER LOC NOT COMPARED**			**217**			
29									***TOTAL LINES NOT APPLICABLE***			***358***			
30									**TOTAL LOC COMPARED MANUALLY**			**72**			

Figure X-1: Line-by-line compare of d_playbox to copyright p_deposit

56. In general, upon manual review, differences detected by difflib tool as part of the automated script, were found to be caused by the following:

a. Differences resulting from differently partitioned source code, e.g. JetBrains used a separate file for each C# class, whereas the Plaintiff's developer had registered the Copyright using three files which contained all the C# classes.

b. Differences because of unmatched synonyms, where the complexity of describing the synonym precluded its inclusion in the scripted comparison. These categories of excluded synonym detection included, but were not limited to:

   i. Multi-line lambda expressions

   ii. Swapped and negated boolean conditionals

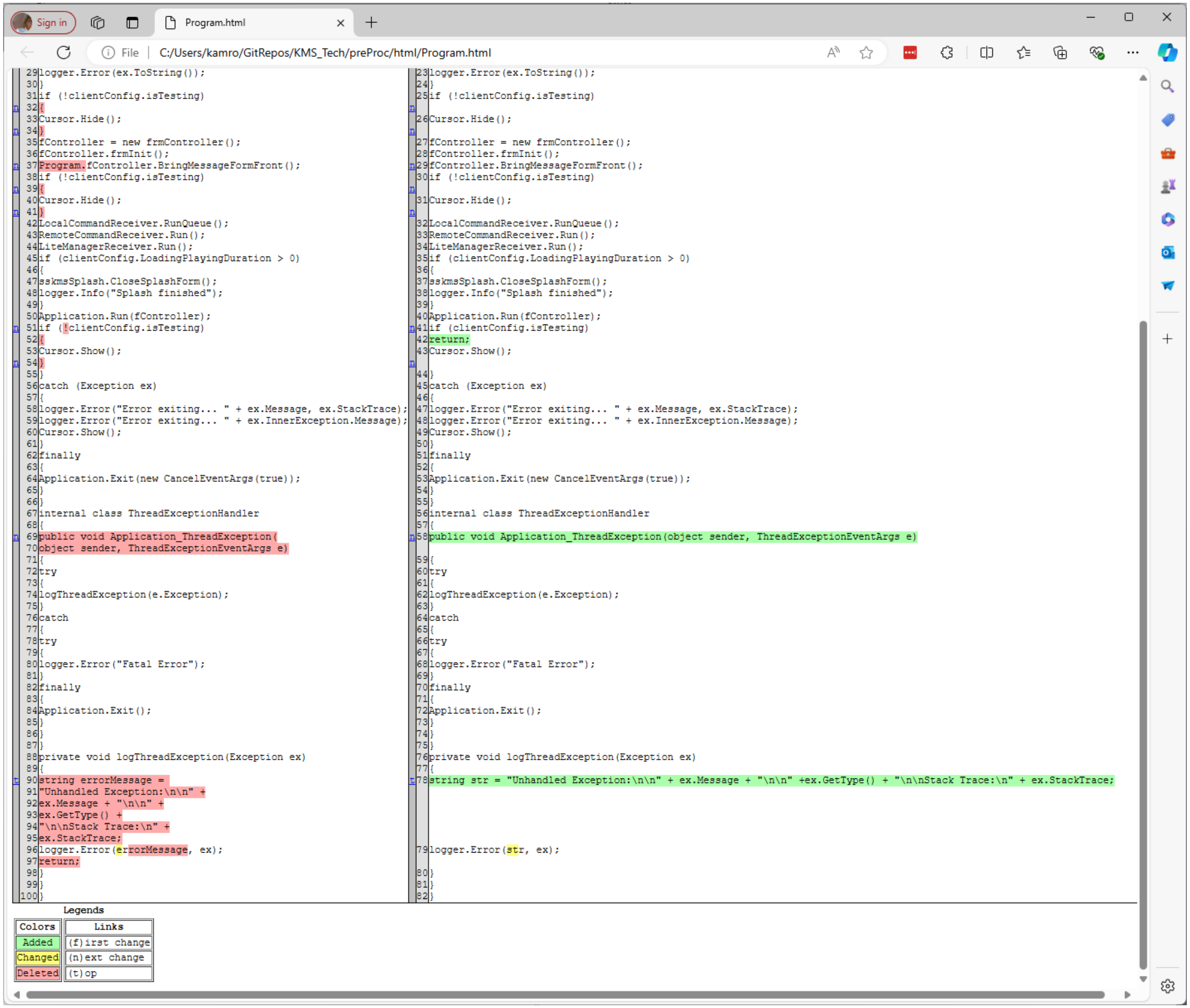
   iii. Multi-line continuation

c. Differences because of trivial source code revisions, such as bug fixes that may have occurred subsequent to the Copyright registration.

57. An excerpt visual example[13] of the scripted difflib comparison of d_playbox and p_deposit is the screenshot of a comparison of a few lines of the class named "Program". (*See Figure X-2, below, with differences highlighted in red and green.*)

58. The complete HTML comparison output can be found in the accompanying production material [*see the files, one for each file of the comparison found in the accompanying production material at "../KMSTech_ProductionMaterial/analysisAndResults/html/*"*]. Viewing these files one can easily see the basis for my conclusion that the d_playbox could only have been created using source code substantially copied from the Plaintiff's Copyright Deposit.

[13] This example is the HTML output obtained using the scripted difflib [*see supra, footnote 5*]



```
logger.Error(ex.ToString());
}
if (!clientConfig.isTesting)
{
Cursor.Hide();
}
fController = new frmController();
fController.frmInit();
Program.fController.BringMessageFormFront();
if (!clientConfig.isTesting)
{
Cursor.Hide();
}
LocalCommandReceiver.RunQueue();
RemoteCommandReceiver.Run();
LiteManagerReceiver.Run();
if (clientConfig.LoadingPlayingDuration > 0)
{
sskmsSplash.CloseSplashForm();
logger.Info("Splash finished");
}
Application.Run(fController);
if (!clientConfig.isTesting)
{
Cursor.Show();
}
}
catch (Exception ex)
{
logger.Error("Error exiting... " + ex.Message, ex.StackTrace);
logger.Error("Error exiting... " + ex.InnerException.Message);
Cursor.Show();
}
finally
{
Application.Exit(new CancelEventArgs(true));
}
}
internal class ThreadExceptionHandler
{
public void Application_ThreadException(
object sender, ThreadExceptionEventArgs e)
{
try
{
logThreadException(e.Exception);
}
catch
{
try
{
logger.Error("Fatal Error");
}
finally
{
Application.Exit();
}
}
}
private void logThreadException(Exception ex)
{
string errorMessage =
"Unhandled Exception:\n\n" +
ex.Message + "\n\n" +
ex.GetType() +
"\n\nStack Trace:\n" +
ex.StackTrace;
logger.Error(errorMessage, ex);
return;
}
}
}
```

```
logger.Error(ex.ToString());
}
if (!clientConfig.isTesting)

Cursor.Hide();

fController = new frmController();
fController.frmInit();
fController.BringMessageFormFront();
if (!clientConfig.isTesting)

Cursor.Hide();

LocalCommandReceiver.RunQueue();
RemoteCommandReceiver.Run();
LiteManagerReceiver.Run();
if (clientConfig.LoadingPlayingDuration > 0)
{
sskmsSplash.CloseSplashForm();
logger.Info("Splash finished");
}
Application.Run(fController);
if (clientConfig.isTesting)
return;
Cursor.Show();

}
catch (Exception ex)
{
logger.Error("Error exiting... " + ex.Message, ex.StackTrace);
logger.Error("Error exiting... " + ex.InnerException.Message);
Cursor.Show();
}
finally
{
Application.Exit(new CancelEventArgs(true));
}
}
internal class ThreadExceptionHandler
{
public void Application_ThreadException(object sender, ThreadExceptionEventArgs e)

{
try
{
logThreadException(e.Exception);
}
catch
{
try
{
logger.Error("Fatal Error");
}
finally
{
Application.Exit();
}
}
}
private void logThreadException(Exception ex)
{
string str = "Unhandled Exception:\n\n" + ex.Message + "\n\n" +ex.GetType() + "\n\nStack Trace:\n" + ex.StackTrace;





logger.Error(str, ex);

}
}
}
```

Legends

Colors	Links
🟩 Added	(f)irst change
🟨 Changed	(n)ext change
🟥 Deleted	(t)op

Figure X-2, Example differences of d_playbox and p_deposit are undetected synonyms

(Left: p_deposit, Right: d_playbox), [*produced by Python difflib*]

XI. METHOD SIGNATURE COMPARISON

59. The following table relates the method-signature elements of a C# program, to elements of the text of a multiple chapter book.

Multi-chapter book element	C# Method-signature element
chapter group heading	class name (*that includes one or more method-signatures*)
chapter title / narrative element	method name
chapter plot outcome	method return value type
character names	method parameter names
character attributes	method parameter types

Figure XI-1 Method-signature Element Correspondence

60. The lines of C# source code that comprise the method-signatures of each class are a concise enumeration of the author's unique creative choices. And the method signatures are included literally in the copyright deposit.

61. Finding similar or identical method signatures of a copyrighted work -- is indicative of copying.

62. A single C# class includes one or more methods, each with a signature exhibiting the developer's creative choices. The Copyright author's selection of meaningful symbolic **class names** and each of the class's one or more **method names**, the **method-parameters**, the **method-parameter names**, the **method-parameter types** and the **method return-types** are literally found in the copyright deposit and can be easily extracted and compared between two bodies of source code (*see below, Figure XI-2 "Annotated Method Signature"*).

63. The method signatures of this extract screenshot example comparison of, "reserveSong" of the class "frmController" are found in the example Figure XI-2 on lines 707 and 476.

64. As can be easily seen, this example method-signature when compared is precisely identical in the two bodies of source code.

65. In general, a method signature includes the following elements (*with the specific example noted*):

i. The method’s class name, i.e. the class name that encapsulates the method, i.e. the class within which the method is defined, *in this example, “frmController”.*

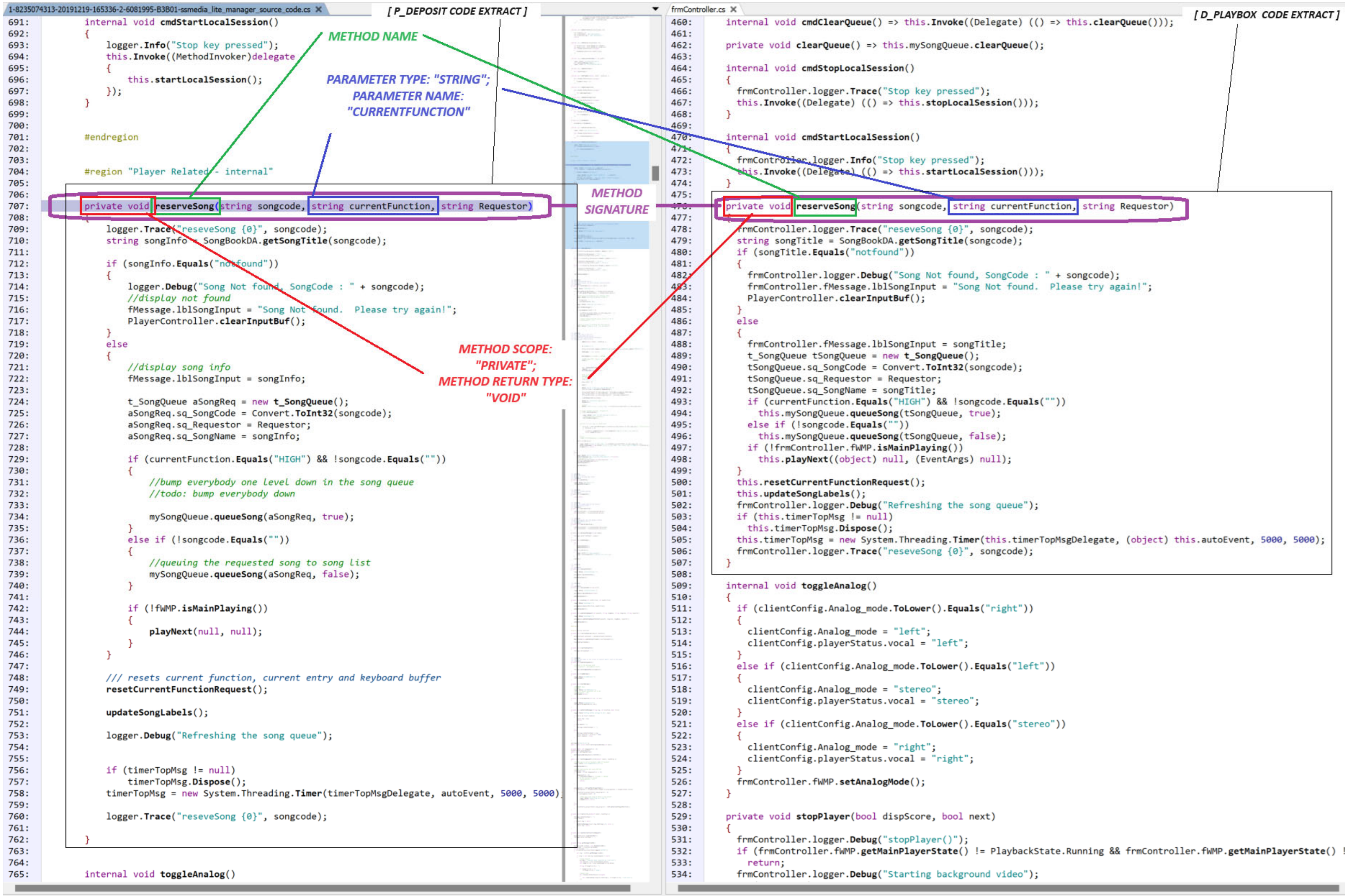
ii. The method name, *in this example, “reserveSong”*

iii. The method return type, *i.e. for “reserveSong”,. “private void”*

iv. The method’s parameters, *i.e. for “reserveSong”, three parameters, (i) “songcode”, (ii) “currentFunction”, and (iii)“Requestor”*

v. The parameter types *i.e. for these three parameters, (i) “string”, (ii) “string”, and (iii) “string”.*

vi. The sequence of the method signature within all the method signatures of the encapsulating class.


1-8235074313-20191219-165336-2-6081995-B3B01-ssmedia_lite_manager_source_code.cs
[P_DEPOSIT CODE EXTRACT]
METHOD NAME
PARAMETER TYPE: "STRING";
PARAMETER NAME:
"CURRENTFUNCTION"
METHOD
SIGNATURE
METHOD SCOPE:
"PRIVATE";
METHOD RETURN TYPE:
"VOID"
frmController.cs
[D_PLAYBOX CODE EXTRACT]

```
        internal void cmdStartLocalSession()
        {
            logger.Info("Stop key pressed");
            this.Invoke((MethodInvoker)delegate
            {
                this.startLocalSession();
            });
        }


        #endregion


        #region "Player Related - internal"

        private void reserveSong(string songcode, string currentFunction, string Requestor)
        {
            logger.Trace("reseveSong {0}", songcode);
            string songInfo = SongBookDA.getSongTitle(songcode);

            if (songInfo.Equals("notfound"))
            {
                logger.Debug("Song Not found, SongCode : " + songcode);
                //display not found
                fMessage.lblSongInput = "Song Not found.  Please try again!";
                PlayerController.clearInputBuf();
            }
            else
            {
                //display song info
                fMessage.lblSongInput = songInfo;

                t_SongQueue aSongReq = new t_SongQueue();
                aSongReq.sq_SongCode = Convert.ToInt32(songcode);
                aSongReq.sq_Requestor = Requestor;
                aSongReq.sq_SongName = songInfo;

                if (currentFunction.Equals("HIGH") && !songcode.Equals(""))
                {
                    //bump everybody one level down in the song queue
                    //todo: bump everybody down

                    mySongQueue.queueSong(aSongReq, true);
                }
                else if (!songcode.Equals(""))
                {
                    //queuing the requested song to song list
                    mySongQueue.queueSong(aSongReq, false);
                }

                if (!fWMP.isMainPlaying())
                {
                    playNext(null, null);
                }
            }

            /// resets current function, current entry and keyboard buffer
            resetCurrentFunctionRequest();

            updateSongLabels();

            logger.Debug("Refreshing the song queue");


            if (timerTopMsg != null)
                timerTopMsg.Dispose();
            timerTopMsg = new System.Threading.Timer(timerTopMsgDelegate, autoEvent, 5000, 5000)

            logger.Trace("reseveSong {0}", songcode);

        }


        internal void toggleAnalog()
```

```
    internal void cmdClearQueue() => this.Invoke((Delegate) (() => this.clearQueue()));

    private void clearQueue() => this.mySongQueue.clearQueue();

    internal void cmdStopLocalSession()
    {
      frmController.logger.Trace("Stop key pressed");
      this.Invoke((Delegate) (() => this.stopLocalSession()));
    }

    internal void cmdStartLocalSession()
    {
      frmController.logger.Info("Stop key pressed");
      this.Invoke((Delegate) (() => this.startLocalSession()));
    }

    private void reserveSong(string songcode, string currentFunction, string Requestor)
    {
      frmController.logger.Trace("reseveSong {0}", songcode);
      string songTitle = SongBookDA.getSongTitle(songcode);
      if (songTitle.Equals("notfound"))
      {
        frmController.logger.Debug("Song Not found, SongCode : " + songcode);
        frmController.fMessage.lblSongInput = "Song Not found.  Please try again!";
        PlayerController.clearInputBuf();
      }
      else
      {
        frmController.fMessage.lblSongInput = songTitle;
        t_SongQueue tSongQueue = new t_SongQueue();
        tSongQueue.sq_SongCode = Convert.ToInt32(songcode);
        tSongQueue.sq_Requestor = Requestor;
        tSongQueue.sq_SongName = songTitle;
        if (currentFunction.Equals("HIGH") && !songcode.Equals(""))
          this.mySongQueue.queueSong(tSongQueue, true);
        else if (!songcode.Equals(""))
          this.mySongQueue.queueSong(tSongQueue, false);
        if (!frmController.fWMP.isMainPlaying())
          this.playNext((object) null, (EventArgs) null);
      }
      this.resetCurrentFunctionRequest();
      this.updateSongLabels();
      frmController.logger.Debug("Refreshing the song queue");
      if (this.timerTopMsg != null)
        this.timerTopMsg.Dispose();
      this.timerTopMsg = new System.Threading.Timer(this.timerTopMsgDelegate, (object) this.autoEvent, 5000, 5000);
      frmController.logger.Trace("reseveSong {0}", songcode);
    }

    internal void toggleAnalog()
    {
      if (clientConfig.Analog_mode.ToLower().Equals("right"))
      {
        clientConfig.Analog_mode = "left";
        clientConfig.playerStatus.vocal = "left";
      }
      else if (clientConfig.Analog_mode.ToLower().Equals("left"))
      {
        clientConfig.Analog_mode = "stereo";
        clientConfig.playerStatus.vocal = "stereo";
      }
      else if (clientConfig.Analog_mode.ToLower().Equals("stereo"))
      {
        clientConfig.Analog_mode = "right";
        clientConfig.playerStatus.vocal = "right";
      }
      frmController.fWMP.setAnalogMode();
    }

    private void stopPlayer(bool dispScore, bool next)
    {
      frmController.logger.Debug("stopPlayer()");
      if (frmController.fWMP.getMainPlayerState() != PlaybackState.Running && frmController.fWMP.getMainPlayerState() !
        return;
      frmController.logger.Debug("Starting background video");
```

Figure XI-2 Annotated Method Signatures d_playbox and p_deposit, example

66. The conservative 72% identical line-by-line comparison result, counts and percent of identical lines compared establish the "substantial similarity" finding (*see supra section X, and Figure X-1*). This metric, i.e. the line-by-line comparison assumes that each line of source code of p_deposit found in d_playbox contributes uniformly to the conclusion of "substantial similarity" and that each line is equally demonstrative of similarity. I don't believe this assumption is true in source code or in natural language texts.

67. The literal lines of code which are recognizable in the C# syntax as method-signatures[14] embody the lion's share of the author's unique and creative decisions choosing the names of program text entities, the attributes of the named entities and relationships among the named entities of p_deposit.

68. The detailed result of the scripted comparison of method signatures extracted from d_playbox compared to the method signatures of p_deposit is found in Exhibit 2.

69. The files of d_playbox source code are enumerated, in Figure X-1, cells [A2] – [A26]. A sequence number is assigned to each file name for reference between Figure X-1 and Exhibit 2 (*see above, Figure X-1, cells [B2] – [B26], compare to Exhibit 2, cells [A2] – [A256]*).

70. In summary, for approximately 5000 lines of d_playbox source code (see above, *Figure X-1, cells [M27] + [L28] + [L29]*), 252 method-signatures were found (*see detail of Exhibit 2, rows 2-256*).

71. Of these 252 method signatures, only two, i.e., (i) "Dispose" and "(ii) InitializeComponent" of class "sskmsSplash" were not found by the script in p_deposit. Upon detailed investigation these "missing" methods were found to be present in p_deposit, but not

[14] Internal method signatures such as those compared herein are distinguished from a published API (of method signatures), in that internal method signatures are not programmatically accessible by the user of the program. Internal method signatures are not visible or accessible to users of the program. The internal method signatures compared herein do not comprise an API, neither does any API exist, that I am aware of.

detected as result of a limitation of the comparison script[15]. Manual review found these and when compared, they were also found to be identical in d_playbox and p_deposit.

72. Of the 250 methods that were compared by the comparison script, 247 were found to be identical for all properties noted above (*see above, paragraph 65 (i) – (vi)*).

73. Manual review of the three methods not found by the script to be identical, when reviewed manually discovered that the "difference" detected by the comparison script was the use in p_deposit of *upper-case* "O" for "Object", whereas *lower-case* "o" for "object" was found in d_playbox (*see Exhibit 2, rows, 116, 172 and 194*).

74. In summary, by using automated comparison of the method-signatures, and manual review these creative elements were found to be essentially 100% identical between the two source codes. The 100% identical method-signatures finding, strongly amplifies the conservative 72% identical lines of code finding. It provides irrefutable foundation to the conclusion that the d_playbox code is substantially similar to p_deposit and could only have been created using source code copied from p_deposit.

Dated: October 22, 2024

Kenneth J. Amron

Pittsfield, MA

[15] The script failed to detect these two methods of the same containing class name, but when the containing class was defined in two different, non-contiguous file locations using the C# "partial" class designation.

Exhibit 1: Kenneth J. Amron CV (4 pages)

Kenneth J. Amron

545 West End Avenue, #3F
New York, NY 10024

kamron@digital-parts.com
(917) 439-8984

EXPERIENCE

2003 – present

Digital Parts, Inc. -- New York City, NY
Technology Consultant: I provide expert technology consulting (*software projects enumerated inline*) *and also* support for patent, copyright and trade secret cases (*see* '**Litigation Support'** *section below*). Recent enterprise software consulting projects have included software design, development and dev-ops projects: ***US-wide real estate transaction data digital media retrieval system***, [for Smarter Agent, (2017-2023)]; ***Airport Risk Planning -- platform for US airports*** [for FAA Transportation Research Board, (2012/3)]; and a follow-up project, ***Airport Emergency Planning platform*** targeted to Node Webkit, [also for FAA TRB, (2017/8)]; ***Platform for video event integration*** *(e.g. sensor enhanced and POS, video-events, etc* [for Time Warner Cable, Atlanta Hartsfield International Airport, Banco Santander and others (2003-2011)].

2001 - 2003

IDT Technology Ventures, L.L.C. -- New York City, NY
Founder, Technology Lead: I provided technology consulting to IDT Corporation, leading a rapid prototype development group and advising IDT's CEO regarding investment opportunities. ***Media ventures*** included: TV.TV, A web-based CDN innovation hosting a digital media marketplace supporting: transaction settlement, content acquisition, digital rights management. ***Financial service initiatives*** included: Caller-paid purchasing – leveraging IDT's net2Phone and pre-paid phone service for E-commerce.

1999 - 2001

Arbinet -- New York City, NY
CTO / VP Software Development: I directed software development and technology strategy, creating an operational ***global trading system for telecom "minutes"***— the key ingredient towards Arbinet's IPO success and acquisition. Arbinet's *AGCN Exchange* advertised international voice communication buy/sell offers by wholesale telecom providers, matching trades to originate or terminate voice communications based on capacity, quality and market. At the daily close, the system would re-route capacity, calculate wholesale settlement and generate end-user billing.

1981 - 1999

Dun & Bradstreet, DunsGate -- New York City, NY
Global Architect**:** I unified **D&B's commercial credit technology offerings**, expanding services to over 70 countries deploying to European, Asia Pacific and US data centers.
Director, Software Development**:** Managed, mentored and incentivized technical staff, building an easily scaled, messaging gateway, supporting internationally diverse D&B financial services for IVR, FAX, network and desktop access.
Lead Developer: Built network concentrator, developed gateway communication protocols and application framework supporting remote interactive voice response platform, deployed internationally.
Systems Engineer: Device drivers and applications for custom A/D and D/A hardware, CSMA/CD, SYNC and ASYNC gateway communications.

1979 - 1981

Advanced Computer Techniques -- New York City, NY,
Key Engineer for ***compiler and operating system initiatives*** *sold to: NASA and DARCOM****.*** *Developed run-time interfaces and code-gen for DARCOM's portable OS / compiler initiative, for Pascal, Fortran, Cobol runtime and inter-process communication.*

1977 - 1979

Burroughs Corporation, Federal and Special Systems Group -- Paoli, Pennsylvania
Systems Engineer for ***Operating System of BSP supercomputer****. Developed file-subsystem and FORTRAN run-time. BSP was competitor of the CRAY-1. Developed the OS Master Control Program on simulator and migrated to prototype hardware.*

SOFTWARE SKILLS

Enterprise software
Product development
Distributed computing
Media processing
Signal processing
System integration & deployment
Real-time data communications
Operations Integration
Team practices
Requirements synthesis
Monitoring, logging, metrics
Team leadership & mentoring
Software Development
C/C++

Python
Java
Javascript
C#/VB/C++/CLI
COBOL
SQL/PL-SQL
Linux/Ubuntu/WSL
NET/Visual Studio/Installshield
Salesforce/Apex/Force.com
Oracle Application Framework
CX_Oracle (Python, C++)
Oracle SQL Developer
AWS – EC2, S3
Google Cloud - GCP

Datadog
Postman
Mozilla/XUL/Firebug
NodeJS/Node-Webkit
XULRunner
Silverlight
Logmatic
Javascript/jQuery/Ajax/JSON
HTML/XHTML/CSS/HTML5
XML/XSLT/XSD/XPath/SOAP
GIT / SVN / CVS
Sybase
DB2
OpenGL, WebGL

1982 - 1986 **New York University, Courant Institute of Mathematical Sciences**
New York City, NY,
Master of Science – Applied Mathematics / Computer and Information Science

1973 - 1977 **State University of New York at Albany**
Albany, NY,
Bachelor of Science - Applied Mathematics / Computer Science (Magna Cum Laude)

LITIGATION SUPPORT

4/2022 – Current: Consultant to Markit Group Ltd., *Capelogic, Inc. v. Markit Group Limited,* Case 3:22-cv-00085-ZNQ-RLS. Technology and code review for a trade secret / contract dispute. (Gibbons, P.C., Samuel Portnoy, Kate Janukowicz).

5/2021 – Current: Consultant to Kewazinga*, Kewazinga Corp. v. Google.,* Case No. 1:20-cv-1106-LGS. Source code review of Google Maps, Streetview feature and expert report in support of a patent dispute. (Brown & Rudnick, contact Ian G. DiBernardo).

5/2019 – 3/2020: Consultant to Kewazinga*, Kewazinga Corp. v. Microsoft Corp.,* No. 1:18-cv-4500-GHW. Source code review of Microsoft Bing Maps, Streetside feature in support of a patent dispute. (Stroock & Lavan, contact Kenneth Stein).

6/2018 - 2020: Consultant to Ultimate Software Group, Inc, re: *Master Tax LLC v. Ultimate Software*, Case No 2:18-cv-01463. Litigation support, reviewing both parties' software and claims in connection with a trade-secret dispute. (Stroock & Lavan, contact Jason Sobel).

1/2018: Consultant to US Phillips, Corp, *Philips v. HTC*, Case No 15-1126-GMS for code review of AMR-WB CODEC and FLACExtractor signal processing. I reviewed Android code and drafted infringement contentions. (Fitzpatrick, contact Jonathan Sharret)

4/2016 – 11/2017: Consultant to Broadsoft, Inc, *Blinkmind. v Broadsoft,* Case No. 2015-16576 in The District Court of Harris County, TX. I reviewed three different parties' multi-point video conferencing software. I performed a three-way, exhaustive GIT/SVN version comparison to quantify and report on evidence of similarities in the context of allegations of copying and trade-secret misappropriation. (Cooley, LLP, contact Peter Yi)

8/2014-8/2015: Consultant to Art+Com Innovationpool, GmbH in *Art+Com Innovationpool GmbH., v. Google Inc.*, Case No 1:14-cv-00217-UNA to perform code review of Google Earth. (Baker Botts, Ryan Pinckney)

5/2014 -11/2016: Consultant to CSX in *Integrated Data Communication Systems, Inc. v. CSX Intermodal Terminals, Inc.*, Case No 2:13-cv-00994. I performed review of both parties' source code and provided fact finding and expert report regarding a trade-secret / copyright dispute. (Gibbons, Christopher Walsh)

3/2014: Consultant to Jobscience in *Jobscience, Inc. v. CVPartners, Inc. et al*, Case No. 3:2013cv04519, California Northern District. Engaged to perform a code review of both party's software in context of a trade-secret and copyright dispute. I provided expert report to the court regarding evidence of copying. (CounselForce, Patrick Terry)

11/2013: Consultant to Opentext in *OpenText S.A. v. Alfresco Software LTD*, Case No. 2:13CV320AWA/LRL; Engaged by Opentext to (i) host, (ii) make operable and (iii) review code of a collaborative software product, in support of infringement contentions. (Cooley)

5/2013: Consultant to Siemens Corporation in *ROY-G-BIV Corp. v. Siemens Corporation et al.*, Case No. 6:11-cv-00624-LED (E.D.Tex.) Engaged regarding a Motion-To-Compel: "Production and Deployment of Complete Development Environment Infrastructure" as defendant's expert regarding system builds and industry practices (i.e., Automation Systems, Operator Monitoring Systems, Industrial Controls, Product Lifecycle Management, Microsoft COM and .NET, Visual Studio) (Kirkland, James Medek)

2/2013: Consultant to Tomita Technologies in *Tomita Technologies USA LLC et al. v. Nintendo Co. Ltd. et al.*, case number 1:11-cv-04256, in the U.S. District Court for the Southern District of New York. On behalf of Plaintiff, performed code review (C/C++) of 3D-imaging/gaming and stereoscopic camera for plaintiff's '664 patent; I provided declaration, expert report, was deposed on two occasions as software expert and testified at trial. (Stroock & Lavan, Ken Stein)

1/2013: Consultant to Daedalus Technology Group, Inc. in *Softview v. Huawei*: Reviewed open source to locate Webkit pinch-scale feature.

8/2012: Consultant to Sony in *HumanEyes Technologies, Ltd. V. Sony Electronics, Inc. et al*, case number 1:12-cv-00398, Delaware District Court. The case was a patent dispute regarding 3D sweep panorama imaging. (Kenyon, Michael Sander)

5/2012: Consultant to Daedalus Technology Group, Inc. in support of a patent dispute. Engaged to develop a benchmark to exercise network switching device features and measure corresponding performance -- assembled legacy hardware, installed boot OS and integration environment, designed, coded and reported on benchmark (LZW compression, C/C++/Ubuntu).

2/2012: Consultant to Daedalus Technology Group, Inc. in defense of patents of a university plaintiff client, I reviewed alleged infringing manufacturer's whitepapers and documentation to draft contentions in defense of patents regarding malware detection and network intrusion.

6/2011: Consultant to Daedalus Technology Group, Inc. in *Apple Inc. v. Nokia Corporation and High Tech Computer Corp*. The matter involved a patent dispute for which I developed a software harness and utilities utilizing FFTW and a LLS approximation to exercise embedded voice 3GPP AMR CODEC and analyze the power spectrum, voice activity detection and bandwidth modes (C/C++).

1/2011: Consultant to Bidz.com, Inc., in *Soverain Software LLC, v. J.C. Penney, et al*, case number 6:09-cv-00274 (E.D. Texas). For Bidz.com, in this patent dispute, I analyzed protocol traces focusing on authentication, sessionID and eCommerce protocols and provided the Bidz non-infringement report content. (Seldon & Scillieri, John Scilleri)

6/2010: Consultant to Daedalus Technology Group in *Spread Spectrum Screening, LLC v. Eastman Kodak Company,* case number 10-CV-1101 for the Norther District of Illinois. For the defendant I developed utilities (Java, AWT) for 2D-Fourier spectral analysis of digital image library.

EXPERT TESTIMONY

3/2022: For Kewazing in *Kewazinga v. Google*, deposed regarding expert report related to review of the Google Maps Streetview feature.

8/2015: For Tomita in *Tomita Technologies USA LLC et al. v. Nintendo Co.*, testified at trial regarding findings related to review of the Nintendo 3DS software in light of US Federal Circuit Court of Appeal, decision.

4/2015: For Tomita in *Tomita Technologies USA LLC et al. v. Nintendo Co.*, deposed regarding expert report related to review of the Nintendo 3DS software in light of US Federal Circuit Court of Appeal, decision dated, December 8, 2014.

6/2013: For Siemens in *ROY-G-BIV Corp. v. Siemens Corporation et al,* deposed regarding findings related to plaintiff's "Motion to Compel Production and Deployment".

3/2013: For Tomita in *Tomita Technologies USA LLC et al. v. Nintendo Co.*, deposed regarding review of the Nintendo 3DS software.

Exhibit 2: Method-signature scripted comparison detailed results (5 pages)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	sort Seq	D class	D class NOT found in P	D class found in P	D method	D method NOT found in P	D method found in P	identical method signature	not identical method signature	D sig	D method lineNo start	P method lineNo start - if found	P fName - if found	P sig (if found but not identical)
2	3	ucSongEntry	0	1	ucSongEntry	0	1	1	0	public ucSongEntry()	17	2468	playbox_souece_code.txt	
3	3	ucSongEntry	0	1	setSongInfo	0	1	1	0	public void setSongInfo(t_SongQueue aSong)	22	2473	playbox_souece_code.txt	
4	3	ucSongEntry	0	1	resetSongInfo	0	1	1	0	public void resetSongInfo()	44	2503	playbox_souece_code.txt	
5	3	ucSongEntry	0	1	Dispose	0	1	1	0	protected override void Dispose(bool disposing)	50	2516	playbox_souece_code.txt	
6	3	ucSongEntry	0	1	InitializeComponent	0	1	1	0	private void InitializeComponent()	56	2525	playbox_souece_code.txt	
7	4	PlayerAudioUtil	0	1	getVolume	0	1	1	0	public static int getVolume()	15	2763	playbox_souece_code.txt	
8	4	PlayerAudioUtil	0	1	setVolume	0	1	1	0	public static int setVolume(int new_volume)	23	2771	playbox_souece_code.txt	
9	4	PlayerAudioUtil	0	1	setInitialVolume	0	1	1	0	public static int setInitialVolume()	36	2788	playbox_souece_code.txt	
10	5	SSMediaPlayerCommon	0	1	OnGraphEvent	0	1	1	0	public void OnGraphEvent(EventCode eventCode, IntPtr param1, IntPtr param2)	38	2211	ssmedia_lite_manager_source_code.txt	
11	5	SSMediaPlayerCommon	0	1	GetCurrentPlayerPosition	0	1	1	0	public int GetCurrentPlayerPosition()	66	2239	ssmedia_lite_manager_source_code.txt	
12	5	SSMediaPlayerCommon	0	1	playDemo	0	1	1	0	public bool playDemo(string demoSong)	89	2258	ssmedia_lite_manager_source_code.txt	
13	5	SSMediaPlayerCommon	0	1	Start	0	1	1	0	public virtual void Start(IWin32Window boss)	95	2264	ssmedia_lite_manager_source_code.txt	
14	5	SSMediaPlayerCommon	0	1	setAnalogMode	0	1	1	0	public bool setAnalogMode()	100	2269	ssmedia_lite_manager_source_code.txt	
15	5	SSMediaPlayerCommon	0	1	startMainPlayer	0	1	1	0	public double startMainPlayer(string SongID)	111	2281	ssmedia_lite_manager_source_code.txt	
16	5	SSMediaPlayerCommon	0	1	volumeMute	0	1	1	0	public int volumeMute()	233	2415	ssmedia_lite_manager_source_code.txt	
17	5	SSMediaPlayerCommon	0	1	volumeSet	0	1	1	0	public int volumeSet(int newVolume)	251	2435	ssmedia_lite_manager_source_code.txt	
18	5	SSMediaPlayerCommon	0	1	volumeUp	0	1	1	0	public int volumeUp()	259	2450	ssmedia_lite_manager_source_code.txt	
19	5	SSMediaPlayerCommon	0	1	volumeDown	0	1	1	0	public int volumeDown()	266	2466	ssmedia_lite_manager_source_code.txt	
20	5	SSMediaPlayerCommon	0	1	pitchSet	0	1	1	0	public int pitchSet(int newPitch)	273	2482	ssmedia_lite_manager_source_code.txt	
21	5	SSMediaPlayerCommon	0	1	pitchUp	0	1	1	0	public int pitchUp()	282	2496	ssmedia_lite_manager_source_code.txt	
22	5	SSMediaPlayerCommon	0	1	pitchDown	0	1	1	0	public int pitchDown()	292	2511	ssmedia_lite_manager_source_code.txt	
23	5	SSMediaPlayerCommon	0	1	tempoSet	0	1	1	0	public int tempoSet(int newTempo)	302	2526	ssmedia_lite_manager_source_code.txt	
24	5	SSMediaPlayerCommon	0	1	tempoUp	0	1	1	0	public int tempoUp()	311	2540	ssmedia_lite_manager_source_code.txt	
25	5	SSMediaPlayerCommon	0	1	tempoDown	0	1	1	0	public int tempoDown()	321	2555	ssmedia_lite_manager_source_code.txt	
26	5	SSMediaPlayerCommon	0	1	pauseMainPlayer	0	1	1	0	public bool pauseMainPlayer()	331	2570	ssmedia_lite_manager_source_code.txt	
27	5	SSMediaPlayerCommon	0	1	isMainPlaying	0	1	1	0	public bool isMainPlaying()	336	2575	ssmedia_lite_manager_source_code.txt	
28	5	SSMediaPlayerCommon	0	1	restartMainPlayer	0	1	1	0	public bool restartMainPlayer()	341	2580	ssmedia_lite_manager_source_code.txt	
29	5	SSMediaPlayerCommon	0	1	playBG	0	1	1	0	public virtual bool playBG(bool start)	347	2586	ssmedia_lite_manager_source_code.txt	
30	5	SSMediaPlayerCommon	0	1	stopBG	0	1	1	0	public bool stopBG()	363	2602	ssmedia_lite_manager_source_code.txt	
31	5	SSMediaPlayerCommon	0	1	WndProc	0	1	1	0	protected override void WndProc(ref Message m)	374	2613	ssmedia_lite_manager_source_code.txt	
32	5	SSMediaPlayerCommon	0	1	stopMainPlayer	0	1	1	0	public virtual void stopMainPlayer()	389	2634	ssmedia_lite_manager_source_code.txt	
33	5	SSMediaPlayerCommon	0	1	SetBGPlaying	0	1	1	0	protected virtual void SetBGPlaying()	394	2639	ssmedia_lite_manager_source_code.txt	
34	5	SSMediaPlayerCommon	0	1	SetMainPlaying	0	1	1	0	protected virtual void SetMainPlaying()	398	2643	ssmedia_lite_manager_source_code.txt	
35	5	SSMediaPlayerCommon	0	1	RefreshStop	0	1	1	0	protected virtual void RefreshStop(bool stop)	402	2647	ssmedia_lite_manager_source_code.txt	
36	5	SSMediaPlayerCommon	0	1	initDSPlayers	0	1	1	0	protected virtual void initDSPlayers()	409	2655	ssmedia_lite_manager_source_code.txt	
37	5	SSMediaPlayerCommon	0	1	cleanUp	0	1	1	0	public virtual bool cleanUp()	413	2659	ssmedia_lite_manager_source_code.txt	
38	5	SSMediaPlayerCommon	0	1	InitializeComponent	0	1	1	0	private void InitializeComponent()	430	2676	ssmedia_lite_manager_source_code.txt	
39	6	MediaPlayerImpl	1	0										
40	7	LocalCommandReceiver	0	1	RunQueue	0	1	1	0	public static void RunQueue()	13	3279	playbox_souece_code.txt	
41	7	LocalCommandReceiver	0	1	Readcmd	0	1	1	0	static void Readcmd(int a)	31	3302	playbox_souece_code.txt	
42	7	LocalCommandReceiver	0	1	CommandEvt	0	1	1	0	static void CommandEvt(object sender, RemoteEventArgs e)	42	3315	playbox_souece_code.txt	
43	8	LiteManagerReceiver	0	1	Run	0	1	1	0	public static void Run()	18	3342	playbox_souece_code.txt	
44	8	LiteManagerReceiver	0	1	Read	0	1	1	0	static void Read(TcpClient client)	47	3373	playbox_souece_code.txt	
45	8	LiteManagerReceiver	0	1	CommandEvt	0	1	1	0	static void CommandEvt(object sender, RemoteEventArgs e)	66	3394	playbox_souece_code.txt	
46	8	LiteManagerReceiver	0	1	WriteResponse	0	1	1	0	static void WriteResponse(NetworkStream ns, KaraokeCommand cmd)	71	3399	playbox_souece_code.txt	
47	8	LiteManagerReceiver	0	1	ReceiveCommand	0	1	1	0	static void ReceiveCommand(BinaryReader reader, KaraokeCommand cmd)	94	3422	playbox_souece_code.txt	
48	8	LiteManagerReceiver	0	1	logResponse	0	1	1	0	static void logResponse(KaraokeCommand cmd)	124	3448	playbox_souece_code.txt	
49	9	RemoteCommandReceiver	0	1	Run	0	1	1	0	public static void Run()	18	3481	playbox_souece_code.txt	
50	9	RemoteCommandReceiver	0	1	Read	0	1	1	0	static void Read(TcpClient client)	49	3514	playbox_souece_code.txt	
51	9	RemoteCommandReceiver	0	1	CommandEvt	0	1	1	0	static void CommandEvt(object sender, RemoteEventArgs e)	68	3535	playbox_souece_code.txt	
52	9	RemoteCommandReceiver	0	1	WriteResponse	0	1	1	0	static void WriteResponse(NetworkStream ns, KaraokeCommand cmd)	73	3540	playbox_souece_code.txt	
53	9	RemoteCommandReceiver	0	1	ReceiveCommand	0	1	1	0	static void ReceiveCommand(BinaryReader reader, KaraokeCommand cmd)	96	3563	playbox_souece_code.txt	
54	9	RemoteCommandReceiver	0	1	logResponse	0	1	1	0	static void logResponse(KaraokeCommand cmd)	126	3589	playbox_souece_code.txt	
55	10	ControllerEvent	0	1	CommandHandler	0	1	1	0	public delegate void CommandHandler(object sender, RemoteEventArgs e)	13	2903	playbox_souece_code.txt	
56	11	PlayerController	0	1	clearInputBuf	0	1	1	0	public static void clearInputBuf()	18	2926	playbox_souece_code.txt	
57	11	PlayerController	0	1	redirectKeys	0	1	1	0	public static void redirectKeys(int a)	22	2929	playbox_souece_code.txt	
58	11	PlayerController	0	1	processCommands	0	1	1	0	public static void processCommands(KaraokeCommand cmd)	31	2940	playbox_souece_code.txt	
59	11	PlayerController	0	1	processRemoteKeys	0	1	1	0	public static void processRemoteKeys(KaraokeCommand cmd)	38	2947	playbox_souece_code.txt	
60	11	PlayerController	0	1	processLocalKeys	0	1	1	0	public static void processLocalKeys(KaraokeCommand cmd)	111	3067	playbox_souece_code.txt	
61	14	frmController	0	1	frmController	0	1	1	0	public frmController()	36	35	playbox_souece_code.txt	
62	14	frmController	0	1	frmInit	0	1	1	0	public void frmInit()	43	42	playbox_souece_code.txt	

Class Method Signature Comparison

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
63	14	frmController	0	1	frmController_Load	0	1	1	0	private void frmController_Load(object sender, EventArgs e)	52	51	playbox_souece_code.txt	
64	14	frmController	0	1	readyComponents	0	1	1	0	private void readyComponents()	69	68	playbox_souece_code.txt	
65	14	frmController	0	1	clearComponents	0	1	1	0	private void clearComponents()	108	111	playbox_souece_code.txt	
66	14	frmController	0	1	frmController_KeyDown	0	1	1	0	private void frmController_KeyDown(object sender, KeyEventArgs e)	113	116	playbox_souece_code.txt	
67	14	frmController	0	1	cmdReserveSong	0	1	1	0	internal void cmdReserveSong(string songcode, string currentFunction, KaraokeCommand cmd)	118	123	playbox_souece_code.txt	
68	14	frmController	0	1	cmdMainKey	0	1	1	0	internal void cmdMainKey(KaraokeCommand cmd)	135	140	playbox_souece_code.txt	
69	14	frmController	0	1	cmdBackKey	0	1	1	0	internal void cmdBackKey(KaraokeCommand cmd)	143	147	playbox_souece_code.txt	
70	14	frmController	0	1	cmdResetCurrentFunctionRequest	0	1	1	0	internal void cmdResetCurrentFunctionRequest()	147	151	playbox_souece_code.txt	
71	14	frmController	0	1	cmdDeleteLastSong	0	1	1	0	internal void cmdDeleteLastSong(KaraokeCommand cmd)	151	158	playbox_souece_code.txt	
72	14	frmController	0	1	cmdDeleteSong	0	1	1	0	internal void cmdDeleteSong(int position)	160	167	playbox_souece_code.txt	
73	14	frmController	0	1	cmdCorrectKey	0	1	1	0	internal void cmdCorrectKey(KaraokeCommand cmd)	169	176	playbox_souece_code.txt	
74	14	frmController	0	1	cmdForward	0	1	1	0	internal void cmdForward(KaraokeCommand cmd)	177	184	playbox_souece_code.txt	
75	14	frmController	0	1	cmdRepeat	0	1	1	0	internal void cmdRepeat(KaraokeCommand cmd)	188	196	playbox_souece_code.txt	
76	14	frmController	0	1	cmdTempoSet	0	1	1	0	internal int cmdTempoSet(KaraokeCommand cmd)	198	206	playbox_souece_code.txt	
77	14	frmController	0	1	cmdTempoUp	0	1	1	0	internal int cmdTempoUp(KaraokeCommand cmd)	210	218	playbox_souece_code.txt	
78	14	frmController	0	1	cmdTempoDown	0	1	1	0	internal int cmdTempoDown(KaraokeCommand cmd)	221	229	playbox_souece_code.txt	
79	14	frmController	0	1	cmdPitchSet	0	1	1	0	internal int cmdPitchSet(KaraokeCommand cmd)	232	240	playbox_souece_code.txt	
80	14	frmController	0	1	cmdPitchUp	0	1	1	0	internal int cmdPitchUp(KaraokeCommand cmd)	244	252	playbox_souece_code.txt	
81	14	frmController	0	1	cmdPitchDown	0	1	1	0	internal int cmdPitchDown(KaraokeCommand cmd)	255	263	playbox_souece_code.txt	
82	14	frmController	0	1	cmdStop	0	1	1	0	internal void cmdStop(KaraokeCommand cmd)	266	274	playbox_souece_code.txt	
83	14	frmController	0	1	cmdTogglePause	0	1	1	0	internal void cmdTogglePause(KaraokeCommand cmd)	275	283	playbox_souece_code.txt	
84	14	frmController	0	1	cmdAnalog	0	1	1	0	internal void cmdAnalog(KaraokeCommand cmd)	284	292	playbox_souece_code.txt	
85	14	frmController	0	1	cmdVolumeMute	0	1	1	0	internal int cmdVolumeMute(KaraokeCommand cmd)	293	301	playbox_souece_code.txt	
86	14	frmController	0	1	cmdVolumeDown	0	1	1	0	internal int cmdVolumeDown(KaraokeCommand cmd)	311	323	playbox_souece_code.txt	
87	14	frmController	0	1	cmdVolumeUp	0	1	1	0	internal int cmdVolumeUp(KaraokeCommand cmd)	323	335	playbox_souece_code.txt	
88	14	frmController	0	1	cmdVolumeSet	0	1	1	0	internal int cmdVolumeSet(KaraokeCommand cmd)	335	347	playbox_souece_code.txt	
89	14	frmController	0	1	cmdRemove	0	1	1	0	internal void cmdRemove(KaraokeCommand cmd)	348	360	playbox_souece_code.txt	
90	14	frmController	0	1	cmdUpdate	0	1	1	0	internal void cmdUpdate(KaraokeCommand cmd)	356	368	playbox_souece_code.txt	
91	14	frmController	0	1	cmdQuery	0	1	1	0	internal void cmdQuery(KaraokeCommand cmd)	369	381	playbox_souece_code.txt	
92	14	frmController	0	1	cmdAssistance	0	1	1	0	internal void cmdAssistance(KaraokeCommand cmd)	418	442	playbox_souece_code.txt	
93	14	frmController	0	1	cmdMove	0	1	1	0	internal void cmdMove(KaraokeCommand cmd)	424	449	playbox_souece_code.txt	
94	14	frmController	0	1	cmdSetInputMessage	0	1	1	0	internal void cmdSetInputMessage(string keybuf)	433	458	playbox_souece_code.txt	
95	14	frmController	0	1	cmdResetPlayer	0	1	1	0	internal void cmdResetPlayer()	439	464	playbox_souece_code.txt	
96	14	frmController	0	1	cmdPlayNext	0	1	1	0	internal void cmdPlayNext(object sender, EventArgs e)	443	468	playbox_souece_code.txt	
97	14	frmController	0	1	cmdMaxPlayerSize	0	1	1	0	internal void cmdMaxPlayerSize()	447	475	playbox_souece_code.txt	
98	14	frmController	0	1	cmdReducePlayerSize	0	1	1	0	internal void cmdReducePlayerSize()	451	482	playbox_souece_code.txt	
99	14	frmController	0	1	cmdClearQueue	0	1	1	0	internal void cmdClearQueue()	455	489	playbox_souece_code.txt	
100	14	frmController	0	1	clearQueue	0	1	1	0	private void clearQueue()	459	496	playbox_souece_code.txt	
101	14	frmController	0	1	cmdStopLocalSession	0	1	1	0	internal void cmdStopLocalSession()	463	500	playbox_souece_code.txt	
102	14	frmController	0	1	cmdStartLocalSession	0	1	1	0	internal void cmdStartLocalSession()	471	508	playbox_souece_code.txt	
103	14	frmController	0	1	reserveSong	0	1	1	0	private void reserveSong(string songcode, string currentFunction, string Requestor)	479	518	playbox_souece_code.txt	
104	14	frmController	0	1	toggleAnalog	0	1	1	0	internal void toggleAnalog()	511	556	playbox_souece_code.txt	
105	14	frmController	0	1	stopPlayer	0	1	1	0	private void stopPlayer(bool dispScore, bool next)	530	575	playbox_souece_code.txt	
106	14	frmController	0	1	playNext	0	1	1	0	private void playNext(object sender, EventArgs e)	548	595	playbox_souece_code.txt	
107	14	frmController	0	1	pausePlay	0	1	1	0	private void pausePlay()	602	659	playbox_souece_code.txt	
108	14	frmController	0	1	playRestart	0	1	1	0	private bool playRestart()	607	664	playbox_souece_code.txt	
109	14	frmController	0	1	maxPlayerSize	0	1	1	0	private void maxPlayerSize()	611	668	playbox_souece_code.txt	
110	14	frmController	0	1	reducePlayerSize	0	1	1	0	private void reducePlayerSize()	616	673	playbox_souece_code.txt	
111	14	frmController	0	1	setInputMessage	0	1	1	0	private void setInputMessage(string keybuf)	621	678	playbox_souece_code.txt	
112	14	frmController	0	1	resetPlayer	0	1	1	0	private void resetPlayer()	625	682	playbox_souece_code.txt	
113	14	frmController	0	1	removeLastSong	0	1	1	0	private void removeLastSong()	639	696	playbox_souece_code.txt	
114	14	frmController	0	1	removeSongAt	0	1	1	0	private void removeSongAt(int position)	645	702	playbox_souece_code.txt	
115	14	frmController	0	1	moveSong	0	1	1	0	private void moveSong(int oldPosition, int newPosition)	651	708	playbox_souece_code.txt	
116	14	frmController	0	1	timerTopMsgClear	0	1	0	1	private void timerTopMsgClear(object stateInfo)	667	722	playbox_souece_code.txt	private void timerTopMsgClear(Object stateInfo)
117	14	frmController	0	1	clearTopSongInfo	0	1	1	0	private void clearTopSongInfo()	672	728	playbox_souece_code.txt	
118	14	frmController	0	1	updateSongLabels	0	1	1	0	private void updateSongLabels()	676	732	playbox_souece_code.txt	
119	14	frmController	0	1	hideBGVideo	0	1	1	0	private void hideBGVideo()	680	736	playbox_souece_code.txt	
120	14	frmController	0	1	startBGVideo	0	1	1	0	private void startBGVideo()	685	741	playbox_souece_code.txt	
121	14	frmController	0	1	displayScore	0	1	1	0	private void displayScore(int min, int max)	690	746	playbox_souece_code.txt	
122	14	frmController	0	1	setBottomMessage	0	1	1	0	private void setBottomMessage(string msg, int duration, bool force)	695	751	playbox_souece_code.txt	
123	14	frmController	0	1	setPlayerActive	0	1	1	0	public void setPlayerActive()	713	774	playbox_souece_code.txt	
124	14	frmController	0	1	timerSongQueuePoll_Tick	0	1	1	0	public void timerSongQueuePoll_Tick(object sender, EventArgs e)	717	778	playbox_souece_code.txt	
125	14	frmController	0	1	timer1_Tick_1	0	1	1	0	private void timer1_Tick_1(object sender, EventArgs e)	740	806	playbox_souece_code.txt	
126	14	frmController	0	1	resetCurrentFunctionRequest	0	1	1	0	private void resetCurrentFunctionRequest()	749	816	playbox_souece_code.txt	
127	14	frmController	0	1	getMessageFromPOS	0	1	1	0	private string getMessageFromPOS()	754	821	playbox_souece_code.txt	

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
128	14	frmController	0	1	btnTest_Click	0	1	1	0	private void btnTest_Click(object sender, EventArgs e)	780	852	playbox_souece_code.txt	
129	14	frmController	0	1	checkSessionOpen	0	1	1	0	private bool checkSessionOpen()	784	859	playbox_souece_code.txt	
130	14	frmController	0	1	startLocalSession	0	1	1	0	private void startLocalSession()	823	907	playbox_souece_code.txt	
131	14	frmController	0	1	stopLocalSession	0	1	1	0	private void stopLocalSession()	831	916	playbox_souece_code.txt	
132	14	frmController	0	1	exitSSKMS	0	1	1	0	internal void exitSSKMS()	842	930	playbox_souece_code.txt	
133	14	frmController	0	1	isMainPlaying	0	1	1	0	internal bool isMainPlaying()	860	950	playbox_souece_code.txt	
134	14	frmController	0	1	isMediaPlaying	0	1	1	0	internal bool isMediaPlaying()	864	954	playbox_souece_code.txt	
135	14	frmController	0	1	isMediaPaused	0	1	1	0	internal bool isMediaPaused()	868	962	playbox_souece_code.txt	
136	14	frmController	0	1	BringMessageFormFront	0	1	1	0	internal void BringMessageFormFront()	872	970	playbox_souece_code.txt	
137	14	frmController	0	1	BringMessageFormTop	0	1	1	0	internal void BringMessageFormTop()	876	977	playbox_souece_code.txt	
138	14	frmController	0	1	SongQueueTimerStart	0	1	1	0	internal void SongQueueTimerStart()	880	984	playbox_souece_code.txt	
139	14	frmController	0	1	Dispose	0	1	1	0	protected override void Dispose(bool disposing)	885	999	playbox_souece_code.txt	
140	14	frmController	0	1	InitializeComponent	0	1	1	0	private void InitializeComponent()	891	1008	playbox_souece_code.txt	
141	14	frmController	0	1	PlayNextHandler	0	1	1	0	public delegate void PlayNextHandler(object sender, EventArgs e)	917	23	playbox_souece_code.txt	
142	14	frmController	0	1	updateSongInfoLabel	0	1	1	0	private delegate void updateSongInfoLabel()	918	27	playbox_souece_code.txt	
143	15	frmRemoteController	0	1	frmRemoteController	0	1	1	0	public frmRemoteController()	77	1069	playbox_souece_code.txt	
144	15	frmRemoteController	0	1	frmController_Load	0	1	1	0	private void frmController_Load(object sender, EventArgs e)	82	1074	playbox_souece_code.txt	
145	15	frmRemoteController	0	1	button1_Click	0	1	1	0	private void button1_Click(object sender, EventArgs e)	85	1078	playbox_souece_code.txt	
146	15	frmRemoteController	0	1	button2_Click	0	1	1	0	private void button2_Click(object sender, EventArgs e)	88	1081	playbox_souece_code.txt	
147	15	frmRemoteController	0	1	button3_Click	0	1	1	0	private void button3_Click(object sender, EventArgs e)	91	1084	playbox_souece_code.txt	
148	15	frmRemoteController	0	1	button4_Click	0	1	1	0	private void button4_Click(object sender, EventArgs e)	94	1087	playbox_souece_code.txt	
149	15	frmRemoteController	0	1	button5_Click	0	1	1	0	private void button5_Click(object sender, EventArgs e)	97	1090	playbox_souece_code.txt	
150	15	frmRemoteController	0	1	button7_Click	0	1	1	0	private void button7_Click(object sender, EventArgs e)	100	1093	playbox_souece_code.txt	
151	15	frmRemoteController	0	1	button8_Click	0	1	1	0	private void button8_Click(object sender, EventArgs e)	103	1096	playbox_souece_code.txt	
152	15	frmRemoteController	0	1	button9_Click	0	1	1	0	private void button9_Click(object sender, EventArgs e)	106	1099	playbox_souece_code.txt	
153	15	frmRemoteController	0	1	button6_Click	0	1	1	0	private void button6_Click(object sender, EventArgs e)	109	1102	playbox_souece_code.txt	
154	15	frmRemoteController	0	1	button10_Click	0	1	1	0	private void button10_Click(object sender, EventArgs e)	112	1105	playbox_souece_code.txt	
155	15	frmRemoteController	0	1	button11_Click	0	1	1	0	private void button11_Click(object sender, EventArgs e)	115	1108	playbox_souece_code.txt	
156	15	frmRemoteController	0	1	btnEnter_Click	0	1	1	0	private void btnEnter_Click(object sender, EventArgs e)	118	1111	playbox_souece_code.txt	
157	15	frmRemoteController	0	1	btnExit_Click_1	0	1	1	0	private void btnExit_Click_1(object sender, EventArgs e)	122	1115	playbox_souece_code.txt	
158	15	frmRemoteController	0	1	buttonKey_Click	0	1	1	0	private void buttonKey_Click(object sender, EventArgs e)	126	1119	playbox_souece_code.txt	
159	15	frmRemoteController	0	1	Dispose	0	1	1	0	protected override void Dispose(bool disposing)	130	1132	playbox_souece_code.txt	
160	15	frmRemoteController	0	1	InitializeComponent	0	1	1	0	private void InitializeComponent()	136	1141	playbox_souece_code.txt	
161	15	frmRemoteController	0	1	PlayNextHandler	0	1	1	0	public delegate void PlayNextHandler(object sender, EventArgs e)	651	1059	playbox_souece_code.txt	
162	15	frmRemoteController	0	1	updateSongInfoLabel	0	1	1	0	private delegate void updateSongInfoLabel()	652	1065	playbox_souece_code.txt	
163	16	frmMainMessage	0	1	setUIFeedback	0	1	1	0	public bool setUIFeedback(UIControlMode conmode, int level, string command, bool isInfo)	51	3628	playbox_souece_code.txt	
164	16	frmMainMessage	0	1	frmMainMessage	0	1	1	0	public frmMainMessage()	91	3670	playbox_souece_code.txt	
165	16	frmMainMessage	0	1	frmMainMessage_Load	0	1	1	0	private void frmMainMessage_Load(object sender, EventArgs e)	115	3694	playbox_souece_code.txt	
166	16	frmMainMessage	0	1	SendBackForm	0	1	1	0	public void SendBackForm()	130	3709	playbox_souece_code.txt	
167	16	frmMainMessage	0	1	BringFrontForm	0	1	1	0	public void BringFrontForm()	134	3713	playbox_souece_code.txt	
168	16	frmMainMessage	0	1	BringTopMostForm	0	1	1	0	public void BringTopMostForm()	140	3719	playbox_souece_code.txt	
169	16	frmMainMessage	0	1	setSongQueueMsg	0	1	1	0	public void setSongQueueMsg(SongQueue songEntries)	210	3761	playbox_souece_code.txt	
170	16	frmMainMessage	0	1	setSongCurrent	0	1	1	0	public void setSongCurrent(t_SongQueue songCurrent)	236	3787	playbox_souece_code.txt	
171	16	frmMainMessage	0	1	Start	0	1	1	0	public void Start(IWin32Window boss)	241	3792	playbox_souece_code.txt	
172	16	frmMainMessage	0	1	timerSetBottomLabel	0	1	0	1	private void timerSetBottomLabel(object stateInfo)	246	3797	playbox_souece_code.txt	private void timerSetBottomLabel(Object stateInfo)
173	16	frmMainMessage	0	1	setBottomLabel	0	1	1	0	private void setBottomLabel()	252	3803	playbox_souece_code.txt	
174	16	frmMainMessage	0	1	getMessageFromPOS	0	1	1	0	public string getMessageFromPOS()	261	3813	playbox_souece_code.txt	
175	16	frmMainMessage	0	1	timerBottomBlink_Tick	0	1	1	0	private void timerBottomBlink_Tick(object sender, EventArgs e)	265	3820	playbox_souece_code.txt	
176	16	frmMainMessage	0	1	displayScore	0	1	1	0	public void displayScore(int min, int max)	278	3836	playbox_souece_code.txt	
177	16	frmMainMessage	0	1	timerScore_Tick	0	1	1	0	private void timerScore_Tick(object sender, EventArgs e)	291	3857	playbox_souece_code.txt	
178	16	frmMainMessage	0	1	timerUIFeedback_Tick	0	1	1	0	private void timerUIFeedback_Tick(object sender, EventArgs e)	299	3865	playbox_souece_code.txt	
179	16	frmMainMessage	0	1	setRoomDurationGuestCnt	0	1	1	0	public void setRoomDurationGuestCnt()	314	3878	playbox_souece_code.txt	
180	16	frmMainMessage	0	1	frmMainMessage_KeyDown	0	1	1	0	private void frmMainMessage_KeyDown(object sender, KeyEventArgs e)	338	3901	playbox_souece_code.txt	
181	16	frmMainMessage	0	1	Dispose	0	1	1	0	protected override void Dispose(bool disposing)	343	3913	playbox_souece_code.txt	
182	16	frmMainMessage	0	1	InitializeComponent	0	1	1	0	private void InitializeComponent()	349	3922	playbox_souece_code.txt	
183	16	frmMainMessage	0	1	updateGuestCountDuration	0	1	1	0	private delegate void updateGuestCountDuration()	643	3622	playbox_souece_code.txt	
184	16	frmMainMessage	0	1	updateBottomMessage	0	1	1	0	private delegate void updateBottomMessage()	644	3624	playbox_souece_code.txt	
185	17	frmMainMessageBar	0	1	setUIFeedback	0	1	1	0	public bool setUIFeedback(UIControlMode conmode, int level, string command, bool isInfo)	39	1855	playbox_souece_code.txt	
186	17	frmMainMessageBar	0	1	frmMainMessageBar	0	1	1	0	public frmMainMessageBar()	79	1896	playbox_souece_code.txt	
187	17	frmMainMessageBar	0	1	frmMainMessageBar_Load	0	1	1	0	private void frmMainMessageBar_Load(object sender, EventArgs e)	92	1909	playbox_souece_code.txt	
188	17	frmMainMessageBar	0	1	SendBackForm	0	1	1	0	public void SendBackForm()	107	1924	playbox_souece_code.txt	
189	17	frmMainMessageBar	0	1	BringFrontForm	0	1	1	0	public void BringFrontForm()	111	1928	playbox_souece_code.txt	
190	17	frmMainMessageBar	0	1	BringTopMostForm	0	1	1	0	public void BringTopMostForm()	117	1934	playbox_souece_code.txt	
191	17	frmMainMessageBar	0	1	setSongQueueMsg	0	1	1	0	public void setSongQueueMsg(SongQueue songEntries)	175	1967	playbox_souece_code.txt	

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
192	17	frmMainMessageBar	0	1	setSongCurrent	0	1	1	0	public void setSongCurrent(t_SongQueue songCurrent)	189	1981	playbox_souece_code.txt	
193	17	frmMainMessageBar	0	1	Start	0	1	1	0	public void Start(IWin32Window boss)	196	1987	playbox_souece_code.txt	
194	17	frmMainMessageBar	0	1	timerSetBottomLabel	0	1	0	1	private void timerSetBottomLabel(object stateInfo)	201	1992	playbox_souece_code.txt	private void timerSetBottomLabel(Object stateInfo)
195	17	frmMainMessageBar	0	1	setBottomLabel	0	1	1	0	private void setBottomLabel()	207	1998	playbox_souece_code.txt	
196	17	frmMainMessageBar	0	1	timerBottomBlink_Tick	0	1	1	0	private void timerBottomBlink_Tick(object sender, EventArgs e)	216	2008	playbox_souece_code.txt	
197	17	frmMainMessageBar	0	1	displayScore	0	1	1	0	public void displayScore(int min, int max)	223	2019	playbox_souece_code.txt	
198	17	frmMainMessageBar	0	1	timerScore_Tick	0	1	1	0	private void timerScore_Tick(object sender, EventArgs e)	237	2041	playbox_souece_code.txt	
199	17	frmMainMessageBar	0	1	timerUIFeedback_Tick	0	1	1	0	private void timerUIFeedback_Tick(object sender, EventArgs e)	246	2050	playbox_souece_code.txt	
200	17	frmMainMessageBar	0	1	setRoomDurationGuestCnt	0	1	1	0	public void setRoomDurationGuestCnt()	261	2063	playbox_souece_code.txt	
201	17	frmMainMessageBar	0	1	frmMainMessageBar_KeyDown	0	1	1	0	private void frmMainMessageBar_KeyDown(object sender, KeyEventArgs e)	264	2067	playbox_souece_code.txt	
202	17	frmMainMessageBar	0	1	Dispose	0	1	1	0	protected override void Dispose(bool disposing)	269	2079	playbox_souece_code.txt	
203	17	frmMainMessageBar	0	1	InitializeComponent	0	1	1	0	private void InitializeComponent()	275	2088	playbox_souece_code.txt	
204	17	frmMainMessageBar	0	1	updateGuestCountDuration	0	1	1	0	private delegate void updateGuestCountDuration()	440	1849	playbox_souece_code.txt	
205	17	frmMainMessageBar	0	1	updateBottomMessage	0	1	1	0	private delegate void updateBottomMessage()	441	1851	playbox_souece_code.txt	
206	19	ucUIFeedBack	0	1	ucUIFeedBack	0	1	1	0	public ucUIFeedBack()	20	2700	ssmedia_lite_manager_source_code.txt	
207	19	ucUIFeedBack	0	1	setMode	0	1	1	0	public bool setMode(UIControlMode conmode, int level, string command)	24	2704	ssmedia_lite_manager_source_code.txt	
208	19	ucUIFeedBack	0	1	setModeNone	0	1	1	0	private void setModeNone()	46	2726	ssmedia_lite_manager_source_code.txt	
209	19	ucUIFeedBack	0	1	setModeOn	0	1	1	0	private void setModeOn()	52	2732	ssmedia_lite_manager_source_code.txt	
210	19	ucUIFeedBack	0	1	setModePitch	0	1	1	0	private void setModePitch(int level)	58	2738	ssmedia_lite_manager_source_code.txt	
211	19	ucUIFeedBack	0	1	setmodeTempo	0	1	1	0	private void setmodeTempo(int level)	117	2802	ssmedia_lite_manager_source_code.txt	
212	19	ucUIFeedBack	0	1	setModeCommand	0	1	1	0	private void setModeCommand(string cmd)	129	2817	ssmedia_lite_manager_source_code.txt	
213	19	ucUIFeedBack	0	1	setModeVolume	0	1	1	0	private void setModeVolume(int level)	135	2823	ssmedia_lite_manager_source_code.txt	
214	19	ucUIFeedBack	0	1	Dispose	0	1	1	0	protected override void Dispose(bool disposing)	150	2860	ssmedia_lite_manager_source_code.txt	
215	19	ucUIFeedBack	0	1	InitializeComponent	0	1	1	0	private void InitializeComponent()	156	2869	ssmedia_lite_manager_source_code.txt	
216	20	frmWMediaPlayer	0	1	frmWMediaPlayer	0	1	1	0	public frmWMediaPlayer()	19	4267	playbox_souece_code.txt	
217	20	frmWMediaPlayer	0	1	frmWMediaPlayer_Load	0	1	1	0	private void frmWMediaPlayer_Load(object sender, EventArgs e)	25	4273	playbox_souece_code.txt	
218	20	frmWMediaPlayer	0	1	Start	0	1	1	0	public override void Start(IWin32Window boss)	31	4279	playbox_souece_code.txt	
219	20	frmWMediaPlayer	0	1	playBG	0	1	1	0	public override bool playBG(bool start)	36	4284	playbox_souece_code.txt	
220	20	frmWMediaPlayer	0	1	SetBGPlaying	0	1	1	0	protected override void SetBGPlaying()	52	4300	playbox_souece_code.txt	
221	20	frmWMediaPlayer	0	1	SetMainPlaying	0	1	1	0	protected override void SetMainPlaying()	59	4307	playbox_souece_code.txt	
222	20	frmWMediaPlayer	0	1	cleanUp	0	1	1	0	public override bool cleanUp()	66	4314	playbox_souece_code.txt	
223	20	frmWMediaPlayer	0	1	frmWMediaPlayer_KeyDown	0	1	1	0	private void frmWMediaPlayer_KeyDown(object sender, KeyEventArgs e)	81	4329	playbox_souece_code.txt	
224	20	frmWMediaPlayer	0	1	Dispose	0	1	1	0	protected override void Dispose(bool disposing)	86	4341	playbox_souece_code.txt	
225	20	frmWMediaPlayer	0	1	InitializeComponent	0	1	1	0	private void InitializeComponent()	92	4350	playbox_souece_code.txt	
226	21	frmWMediaPlayerBar	0	1	frmWMediaPlayerBar	0	1	1	0	public frmWMediaPlayerBar()	19	2292	playbox_souece_code.txt	
227	21	frmWMediaPlayerBar	0	1	initDSPlayers	0	1	1	0	protected override void initDSPlayers()	24	2297	playbox_souece_code.txt	
228	21	frmWMediaPlayerBar	0	1	frmWMediaPlayerBar_Load	0	1	1	0	private void frmWMediaPlayerBar_Load(object sender, EventArgs e)	29	2302	playbox_souece_code.txt	
229	21	frmWMediaPlayerBar	0	1	Start	0	1	1	0	public override void Start(IWin32Window boss)	36	2309	playbox_souece_code.txt	
230	21	frmWMediaPlayerBar	0	1	playBG	0	1	1	0	public override bool playBG(bool start)	42	2315	playbox_souece_code.txt	
231	21	frmWMediaPlayerBar	0	1	SetBGPlaying	0	1	1	0	protected override void SetBGPlaying()	58	2331	playbox_souece_code.txt	
232	21	frmWMediaPlayerBar	0	1	SetMainPlaying	0	1	1	0	protected override void SetMainPlaying()	65	2338	playbox_souece_code.txt	
233	21	frmWMediaPlayerBar	0	1	cleanUp	0	1	1	0	public override bool cleanUp()	72	2345	playbox_souece_code.txt	
234	21	frmWMediaPlayerBar	0	1	frmWMediaPlayerBar_KeyDown	0	1	1	0	private void frmWMediaPlayerBar_KeyDown(object sender, KeyEventArgs e)	87	2360	playbox_souece_code.txt	
235	21	frmWMediaPlayerBar	0	1	Dispose	0	1	1	0	protected override void Dispose(bool disposing)	92	2372	playbox_souece_code.txt	
236	21	frmWMediaPlayerBar	0	1	InitializeComponent	0	1	1	0	private void InitializeComponent()	98	2381	playbox_souece_code.txt	
237	22	sskmsSplash	0	1	sskmsSplash	0	1	1	0	public sskmsSplash()	24	2814	playbox_souece_code.txt	
238	22	sskmsSplash	0	1	isSplashFinished	0	1	1	0	public static bool isSplashFinished()	28	2823	playbox_souece_code.txt	
239	22	sskmsSplash	0	1	ShowForm	0	1	1	0	public static void ShowForm()	32	2827	playbox_souece_code.txt	
240	22	sskmsSplash	0	1	CloseForm	0	1	1	0	public static void CloseForm()	39	2834	playbox_souece_code.txt	
241	22	sskmsSplash	0	1	ShowSplashScreen	0	1	1	0	public static void ShowSplashScreen()	43	2838	playbox_souece_code.txt	
242	22	sskmsSplash	0	1	CloseSplashForm	0	1	1	0	public static void CloseSplashForm()	52	2847	playbox_souece_code.txt	
243	22	sskmsSplash	0	1	timer1_Tick	0	1	1	0	static void timer1_Tick(object sender, EventArgs e)	56	2851	playbox_souece_code.txt	
244	22	sskmsSplash	0	1	sskmsSplash_Load	0	1	1	0	private void sskmsSplash_Load(object sender, EventArgs e)	81	2877	playbox_souece_code.txt	
245	22	sskmsSplash	0	1	Dispose	1	0							
246	22	sskmsSplash	0	1	InitializeComponent	1	0							
247	23	Program	0	1	Main	0	1	1	0	public static void Main()	18	1733	playbox_souece_code.txt	
248	24	KeyIntercept	0	1	startInterceptKey	0	1	1	0	public static void startInterceptKey(object obj)	17	2702	playbox_souece_code.txt	
249	24	KeyIntercept	0	1	stopInterceptKey	0	1	1	0	public static void stopInterceptKey()	28	2713	playbox_souece_code.txt	
250	24	KeyIntercept	0	1	SetHook	0	1	1	0	static IntPtr SetHook(LowLevelKeyboardProc proc)	33	2718	playbox_souece_code.txt	
251	24	KeyIntercept	0	1	HookCallback	0	1	1	0	static IntPtr HookCallback(int nCode, IntPtr wParam, IntPtr lParam)	41	2728	playbox_souece_code.txt	
252	24	KeyIntercept	0	1	LowLevelKeyboardProc	0	1	1	0	private delegate IntPtr LowLevelKeyboardProc(int nCode, IntPtr wParam, IntPtr lParam)	56	2727	playbox_souece_code.txt	
253	28	Util	0	1	ConvertUsToVolume	0	1	1	0	public static int ConvertUsToVolume(int level)	10	2949	ssmedia_lite_manager_source_code.txt	
254	28	Util	0	1	ConvertVolumeToUs	0	1	1	0	public static int ConvertVolumeToUs(int level)	19	2960	ssmedia_lite_manager_source_code.txt	

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
255	**sort Seq**	**D class**	**D class NOT found in P**	**D class found in P**	**D method**	**D method NOT found in P**	**D method found in P**	**identical method signature**	**not identical method signature**	**D sig**	**D method lineNo start**	**P method lineNo start - if found**	**P fName - if found**	**P sig (if found but not identical)**
256		**TOTALS**	1	252		2	250	247	3					